



**British  
Geological Survey**  
NATURAL ENVIRONMENT RESEARCH COUNCIL

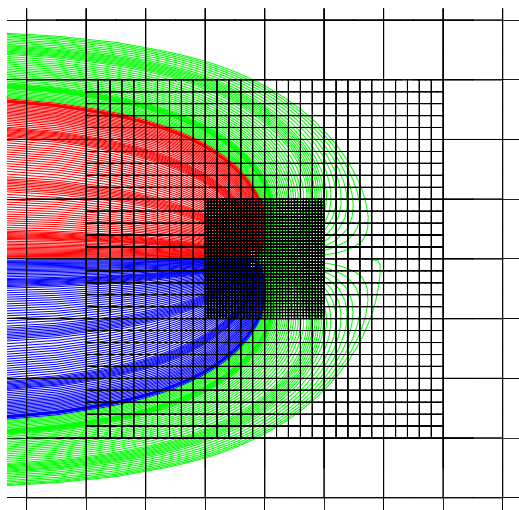


**ENVIRONMENT  
AGENCY**

# Steady-state particle tracking in the object-oriented regional groundwater model ZOOMQ3D

Groundwater Systems & Water Quality Programme  
Commissioned Report CR/02/210C

National Groundwater & Contaminated Land Centre  
Technical Report NC/01/38/2





BRITISH GEOLOGICAL SURVEY  
Commissioned Report CR/02/210C

ENVIRONMENT AGENCY  
National Groundwater & Contaminated Land Centre

Technical Report NC/01/38/2

This report is the result of a study jointly funded by the British Geological Survey's National Groundwater Survey and the Environment Agency's National Groundwater & Contaminated Land Centre in collaboration with The University of Birmingham. No part of this work may be reproduced or transmitted in any form or by any means, or stored in a retrieval system of any nature, without the prior permission of the copyright proprietors. All rights are reserved by the copyright proprietors.

*Disclaimer*

The officers, servants or agents of both the British Geological Survey and the Environment Agency accept no liability whatsoever for loss or damage arising from the interpretation or use of the information, or reliance on the views contained herein.

*Environment Agency  
Dissemination status*

Internal: Release to Regions  
External: Public Domain  
Project No. NC/01/38

*Statement of use*

This document is one of a series of reports describing the development of the groundwater model ZOOMQ3D.

*Key words*

Particle tracking; groundwater model

*Front cover*

Particle tracking in a locally refined grid.

*Bibliographical reference*

JACKSON, C.R. 2002. Steady-state particle tracking in the object-oriented regional groundwater model ZOOMQ3D. *British Geological Survey Commissioned Report*, CR/02/210C.

ISBN 1 85705 989 1

© Environment Agency 2002  
© NERC 2002

# Steady-state particle tracking in the object-oriented regional groundwater model ZOOMQ3D

Author: C.R. Jackson

**Environment Agency Project Manager:**

Paul Hulme

National Groundwater & Contaminated Land Centre

**British Geological Survey Project Manager:**

Dr Andrew Hughes

Groundwater Systems & Water Quality Programme

National Groundwater & Contaminated Land Centre, Solihull 2002  
British Geological Survey, Keyworth, Nottingham 2002

## BRITISH GEOLOGICAL SURVEY

The full range of Survey publications is available from the BGS Sales Desks at Nottingham and Edinburgh; see contact details below or shop online at [www.thebgs.co.uk](http://www.thebgs.co.uk)

The London Information Office maintains a reference collection of BGS publications including maps for consultation.

The Survey publishes an annual catalogue of its maps and other publications; this catalogue is available from any of the BGS Sales Desks.

*The British Geological Survey carries out the geological survey of Great Britain and Northern Ireland (the latter as an agency service for the government of Northern Ireland), and of the surrounding continental shelf, as well as its basic research projects. It also undertakes programmes of British technical aid in geology in developing countries as arranged by the Department for International Development and other agencies.*

*The British Geological Survey is a component body of the Natural Environment Research Council.*

### Keyworth, Nottingham NG12 5GG

☎ 0115-936 3241 Fax 0115-936 3488  
e-mail: [sales@bgs.ac.uk](mailto:sales@bgs.ac.uk)  
[www.bgs.ac.uk](http://www.bgs.ac.uk)  
Shop online at: [www.thebgs.co.uk](http://www.thebgs.co.uk)

### Murchison House, West Mains Road, Edinburgh EH9 3LA

☎ 0131-667 1000 Fax 0131-668 2683  
e-mail: [scotsales@bgs.ac.uk](mailto:scotsales@bgs.ac.uk)

### London Information Office at the Natural History Museum (Earth Galleries), Exhibition Road, South Kensington, London SW7 2DE

☎ 020-7589 4090 Fax 020-7584 8270  
☎ 020-7942 5344/45 email: [bgs\\_london@bgs.ac.uk](mailto:bgs_london@bgs.ac.uk)

### Forde House, Park Five Business Centre, Harrier Way, Sowton, Exeter, Devon EX2 7HU

☎ 01392-445271 Fax 01392-445371

### Geological Survey of Northern Ireland, 20 College Gardens, Belfast BT9 6BS

☎ 028-9066 6595 Fax 028-9066 2835

### Maclean Building, Crowmarsh Gifford, Wallingford, Oxfordshire OX10 8BB

☎ 01491-838800 Fax 01491-692345

### Parent Body

### Natural Environment Research Council, Polaris House, North Star Avenue, Swindon, Wiltshire SN2 1EU

☎ 01793-411500 Fax 01793-411501  
[www.nerc.ac.uk](http://www.nerc.ac.uk)

## ENVIRONMENT AGENCY

The Environment Agency is a non-departmental public body with particular responsibilities for aspects of environmental regulation and management in England and Wales. In discharging these responsibilities the Agency carries out projects both alone and in collaboration with others.

Agency reports and documents are published through a variety of routes. National Groundwater and Contaminated Land Centre publications are obtainable from the address below. Research and Development project reports can be obtained from:

### Environment Agency R&D Dissemination Centre

c/o WRc, Frankland Rd, Swindon, Wilts, SN5 8YF  
☎ 01793 865138 Fax 01793 514562  
[www.wrcplc.co.uk/rdbookshop](http://www.wrcplc.co.uk/rdbookshop)

Other documents are sold through The Stationery Office  
The Publications Centre, PO Box 29, Norwich NR3 1GN  
☎ 0870 600 5522 Fax 0870 600 5533  
e-mail: [customer.services@tso.co.uk](mailto:customer.services@tso.co.uk)

### National Groundwater and Contaminated Land Centre

Olton Court, 10 Warwick Rd  
Olton, Solihull, B92 7HX  
☎ 0121 711 5885 Fax 0121 711 5925  
e-mail: [ngwclc@environment-agency.gov.uk](mailto:ngwclc@environment-agency.gov.uk)

### Environment Agency General Enquiry Line

☎ 0645 333 111

### Environment Agency Regional Offices

#### Anglian

Kingfisher House, Goldhay Way, Orton Goldhay, Peterborough  
PE2 5ZR  
☎ 01733 371811 Fax 01733 231840

#### Midlands

Sapphire East, 550 Streetsbrook Road, Solihull, West Midlands  
B91 1QT  
☎ 0121 711 2324 Fax 0121 711 5824

#### North East

Rivers House, 21 Park Square South, Leeds LS1 2QG  
☎ 0113 244 0191 Fax 0113 246 1889

#### North West

Richard Fairclough House, Knutsford Road, Warrington  
WA4 1HG  
☎ 01925 653999 Fax 01925 415961

#### South West

Manley House, Kestrel Way, Exeter EX2 7LQ  
☎ 01392 444000 Fax 01392 444238

#### Southern

Guildbourne House, Chatsworth Rd, Worthing, Sussex BN11 1LD  
☎ 01903 832000 Fax 01903 821832

#### Thames

Kings Meadow House, Kings Meadow Road, Reading RG1 8DQ  
☎ 0118 953 5000 Fax 0118 950 0388

#### Environment Agency Wales

29 Newport Road, Cardiff CF24 0TP  
☎ 01222 770088 Fax 01222 798555

## Acknowledgements

This work has been undertaken as part of a continuing tripartite collaboration between Dr A. Spink of The University of Birmingham, the Environment Agency, represented by S. Fletcher and P. Hulme of the National Groundwater and Contaminated Land Centre and The British Geological Survey. The project has been co-funded by the Environment Agency's National Groundwater and Contaminated Land Centre and the British Geological Survey.

# Contents

<b>Acknowledgements.....</b>	<b>i</b>
<b>Contents.....</b>	<b>ii</b>
<b>Summary .....</b>	<b>iv</b>
<b>1 Introduction .....</b>	<b>1</b>
<b>2 Particle tracking theory .....</b>	<b>1</b>
2.1 Mathematical background .....	1
2.2 Velocity interpolation .....	2
2.3 Pathline definition.....	4
<b>3 Functionality of the particle tracking code, ZOOPT .....</b>	<b>9</b>
3.1 Introduction .....	9
3.2 Velocity calculation.....	10
3.3 Grid considerations.....	12
3.4 Quasi three-dimensional layering.....	14
3.5 Vertical variation of hydraulic conductivity with depth.....	14
<b>4 Model input and output .....</b>	<b>17</b>
4.1 Model input.....	17
4.2 Model output.....	18
4.3 Running the model.....	18
<b>5 Model testing.....</b>	<b>19</b>
5.1 Introduction .....	19
5.2 Test 1: Analytical solution.....	19
5.3 Test 2: Non-rectangular vertical discretisation.....	21
5.4 Test 3: Quasi 3D layering .....	23
5.5 Test 4: Flow to a pumped well .....	24
5.6 Test 5: Weak sinks.....	27
5.7 Test 6: Particle tracking in VKD layers.....	29
<b>6 Conclusions and recommendations .....</b>	<b>31</b>
<b>7 References .....</b>	<b>32</b>

## FIGURES

Figure 1 Inter-nodal velocity components in the x-direction .....	3
Figure 2 Particle track through a two-dimensional cell .....	5
Figure 3 Cell wall velocity conditions needing consideration during particle tracking .....	6
Figure 4 Intermediate steps taken during fourth-order Runge-Kutta method .....	8
Figure 5 Illustration of a well as a distributed sink at the centre of a finite difference node showing surrounding cell walls in a) three dimensions and b) plan view .....	12
Figure 6 Correction of particle elevation in vertical distorted model layers .....	13
Figure 7 Parameters used to define VKD profiles in ZOOMQ3D .....	15
Figure 8 Illustration of the interpolation of velocity in VKD nodes .....	16
Figure 9 A single pumping well in a uniform flow field, after Bear (1979) .....	19
Figure 10 Comparison of the analytical solution for the shape of a borehole catchment (red) with particle tracks of (a) the MODFLOW model and (b) the ZOOPT model .....	20
Figure 11 Models meshes used to test particle tracking in vertically distorted grids .....	21
Figure 12 Comparison between particle paths in the regular orthogonal grid and the vertically distorted mesh models .....	22
Figure 13 Model grid showing vertical particle tracks through quasi 3D layer .....	23
Figure 14 Model grid used to track particles to a pumped well in a two-layer aquifer .....	24
Figure 15 Particles tracks in plan view and in cross-section for a) the uniform grid and b) the refined grid model .....	25
Figure 16 Particles tracks in 3D isometric view for a) the uniform grid and b) the refined grid model .....	26
Figure 17 Borehole catchments generated around weak sink nodes in Test 5 model using ZOOPT .....	28
Figure 18 Comparison of particle paths in a uniform hydraulic conductivity and a VKD quasi two-layer model .....	30

## TABLES

Table 1 Velocity interpolation and solution methods used in particle tracking codes .....	2
Table 2 ZOOPT input files .....	17
Table 3 ZOOPT output files .....	18
Table 4 Pumping rates for time-variant simulation in Test 5 .....	27

# Summary

This report describes the development of a steady-state particle tracking code for use in conjunction with the object-oriented regional groundwater flow model, ZOOMQ3D (Jackson, 2001). Like the flow model, the particle tracking software, ZOOPT, is written using an object-oriented approach to promote its extensibility and flexibility.

ZOOPT enables the definition of steady-state pathlines in three dimensions. Particles can be tracked in both the forward and reverse directions enabling the rapid definition of borehole catchments, recharge and discharge areas and the visualisation of groundwater flow fields, for example.

Pathlines are defined using the semi-analytical method (Pollock, 1988), however, around particular model features the Runge-Kutta technique is implemented in order to solve some specific problems associated with particle tracking. The problem of particle termination at ‘weak’ sink nodes is solved by the application of the special velocity interpolation scheme presented by Zheng (1994). This approach enables the definition of borehole catchments around wells that induce weak sinks which is not possible with many other widely used particle tracking codes.

ZOOMQ3D incorporates the representation of the vertical variation of hydraulic conductivity with depth (VKD) within finite difference nodes. This has been implemented in the flow model to enable the more accurate description of the variation of hydraulic conductivity in limestone, and particularly Chalk aquifers, in which higher hydraulic conductivities are often associated with the zone of fluctuation of the water table. ZOOPT is fully compatible with VKD models and the application of particle tracking in such inhomogeneous aquifers is a development that is expected to be of significant benefit.

ZOOMQ3D also enables the local refinement of the finite difference grid, for example, around pumping wells. Again, ZOOPT is fully compatible with this model feature and can be used to track particles through such refined meshes.

ZOOPT has been rigorously tested through its comparison with an analytical solution and another particle tracking code and through the visual inspection of pathlines generated using numerous test models. A subset of these tests is presented to illustrate the correct operation of ZOOPT.

Whilst the particle tracking routine currently facilitates the definition of steady-state pathlines only, it enables the rapid visualisation of flow fields, which are based on the node-by-node flows at a specific instant of a time-variant simulation. For example, this capability allows the examination of the changing shape of an approximate borehole catchment over an annual recharge or abstraction cycle.

The next stage in the development of the code will be to implement time-variant particle tracking, which given its structure should be relatively straightforward.



# 1 Introduction

Particle tracking is commonly used to define the pathlines of solute particles under purely advective transport. The technique is often applied for the definition of borehole catchments and associated source protection zones, the identification of recharge and discharge areas and the visualisation of groundwater flow patterns. However, the method also forms the basis of a number of solute transport models, which simulate the effects of hydrodynamic dispersion. Random walk methods (Prickett et al., 1981; Farahmand-Razavi, 1995) and the method of characteristics (Konikow and Bredehoeft, 1978; Zheng, 1990) use particle tracking to describe the advective component of solute transport.

This document describes the development of a particle tracking code, ZOOPT, which is based on the object-oriented regional groundwater flow model, ZOOMQ3D (Jackson, 2001). Particle tracking methods are described prior to the description of the application and testing of the code with relation to the hydrogeological mechanisms incorporated in ZOOMQ3D.

## 2 Particle tracking theory

### 2.1 MATHEMATICAL BACKGROUND

Assuming that fluid density is uniform the pathlines of contaminants under advection alone are governed by the equation

$$\frac{dp}{dt} = v(p, t) \quad (2.1)$$

where,

$p = x \mathbf{i} + y \mathbf{j} + z \mathbf{k}$  is the position vector and,

$v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$  is the seepage velocity vector,

$$v_x = \frac{K_x}{\theta} \frac{\partial h}{\partial x},$$

$K_x$  is the hydraulic conductivity in the x-direction,

$\theta$  is the porosity and,

$h(x, y, z, t)$  is the groundwater head, which is a function of space and time.

The solution of Equation 2.1 for the position of a particle at time,  $t$ , is

$$p(t) = p(t_0) + \int_{t_0}^t v(p, t) dt \quad (2.2)$$

where,

$p(t_0)$  is the initial position of the particle at time  $t_0$ .

The solution of Equation 2.2 requires the evaluation of the velocity field at any given time and position in the model domain. If an exact solution for the velocity field exists then Equation 2.2 can be solved analytically for  $p(t)$ . However, this is generally not the case and Equation 2.2 must then be solved numerically. In a finite difference model velocity components are only known at specific locations in the aquifer, that is, at the position of the cell wall between two adjacent nodes. Consequently, an interpolation scheme must be used to evaluate the velocity field at arbitrary positions and times. This means that an analytical solution to Equation 2.2 cannot be calculated. Furthermore the selection of the interpolation method determines which numerical integration techniques can be used to define the pathline. These considerations are discussed next.

## 2.2 VELOCITY INTERPOLATION

Different velocity interpolation methods have been used in particle tracking codes, of which a number are listed in Table 1. Each velocity interpolation scheme has its advantages and disadvantages, though the selection of a method is often based on the comparison between linear and multi-linear interpolation techniques. The benefit of using simple linear velocity interpolation in each co-ordinate direction is that the technique satisfies finite difference cell-by-cell mass balances (Goode, 1990) and preserves velocity discontinuities at cell boundaries in heterogeneous systems. A disadvantage of the method is that it can produce less realistic pathlines in homogeneous aquifers when compared to higher order interpolation methods, such as bi-linear interpolation. However, a significant benefit of the use of linear velocity interpolation is that it allows Equation 2.2 to be solved using a *semi-analytical* method, which is computationally efficient. The efficiency of the method is discussed in Section 2.3.1.

**Table 1 Velocity interpolation and solution methods used in particle tracking codes**

Particle Tracking Code	Author	Interpolation scheme	Particle movement technique
MOC	Konikow & Bredehoeft (1978)	Bi-linear	Euler integration
RANDOM WALK	Prickett et al. (1981)	Bi-linear	Euler integration
GWPATH	Shafer (1987)	Bi-cubic	Runge-Kutta
MODPATH	Pollock (1989)	Linear	Semi-analytical
PATH3D	Zheng (1989)	Linear	Fourth order Runge-Kutta
FLOWPATH	Franz and Guiger (1990)	Reverse distance	Euler integration
WHPA	Blandford & Huyakorn (1991)	Linear	Semi-analytical or Euler

Linear velocity interpolation is implemented in ZOOPT, the particle tracking code developed here, which is based on the regional groundwater model ZOOMQ3D (Jackson, 2001). As stated above, this enables the analytical solution of the integral in Equation 2.2. The approach also maintains consistency with the finite difference mass balance equations, is generally more accurate than other methods in heterogeneous media and is computationally efficient. These issues are discussed by Zheng and Bennett (1995) who state that simple linear interpolations schemes are generally preferable to multi-linear interpolation schemes. Furthermore, the approach can easily form the basis of a semi-analytical *time-variant* particle tracking technique presented by Lu (1994).

The calculation of the velocity at the cell wall between two nodes is based on the inter-nodal volumetric flow rate calculated by the flow model, ZOOMQ3D. Considering the component in

the x-direction, as illustrated in Figure 1, the velocity at the cell wall at position  $(i-\frac{1}{2}, j)$ , denoted by  $V_{x_{i-\frac{1}{2},j}}$  is

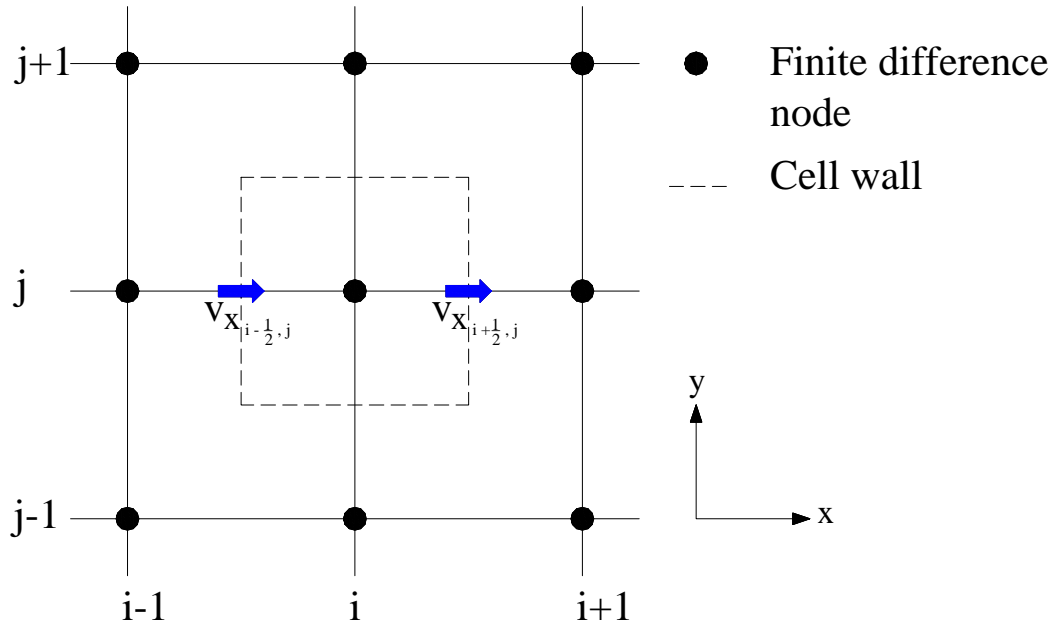
$$V_{x_{i-\frac{1}{2},j}} = \frac{Q_{x_{i-\frac{1}{2},j}}}{\theta \Delta y \Delta z} \text{ (m/day)} \quad (2.3)$$

where,

$Q_{x_{i-\frac{1}{2},j}}$  is the inter-nodal flow rate calculated by the flow model ( $\text{m}^3/\text{day}$ ),

$\theta$  is the aquifer porosity of cell  $(i, j)$ ,

$\Delta y = (y_{j+1} - y_{j-1})/2$  (m) and  $\Delta z$  is the aquifer thickness of cell  $(i, j)$  (m).



**Figure 1 Inter-nodal velocity components in the x-direction**

Given the inter-nodal velocities obtained from Equation 2.3, the x-component of the velocity at any arbitrary location within the cell  $(i, j)$  can be calculated using linear interpolation between the two opposite cell walls. The component of the velocity in the x-direction is calculated at an arbitrary x co-ordinate between  $x_{i-\frac{1}{2}}$  and  $x_{i+\frac{1}{2}}$  using the following linear interpolation equation

$$V_x(x) = A_x \left( x - x_{i-\frac{1}{2}} \right) + V_{x_{i-\frac{1}{2}}} \quad (2.4)$$

where,

$V_x(x)$  is the component of the velocity in the x-direction at  $x$  and

$$A_x = \left( V_{x_{i+\frac{1}{2}}} - V_{x_{i-\frac{1}{2}}} \right) / \left( x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \right) \quad (2.5)$$

Similar equations to Equations 2.3-2.5 are defined in the y and z-directions by replacing the terms, x and i, by the terms y and j, or z and k, respectively. These are used to calculate the remaining two directional components of the velocity vector at any arbitrary position in the model domain.

## 2.3 PATHLINE DEFINITION

### 2.3.1 Semi-analytical technique

If linear interpolation is used to calculate the x, y and z-directional components of the velocity field at a particular position in the finite difference model domain, as described above, then the integral in Equation 2.2 can be solved analytically. Pollock (1988) called this method of particle tracking the ‘semi-analytical’ technique because of the combination of a numerical velocity interpolation routine and an analytical pathline definition procedure. Considering the x-component of the velocity field only, then the equation of the particle track, Equation 2.1, is written

$$\frac{dx}{dt} = V_x \quad \text{or} \quad \frac{1}{V_x} dx = dt \quad (2.6)$$

Substituting Equation 2.4 into 2.6 and integrating between two arbitrary times,  $t_1$  and  $t_2$ , gives

$$\int_{x(t_1)}^{x(t_2)} \frac{1}{A_x \left( x - x_{i-\frac{1}{2}} \right) + V_{x_{i-\frac{1}{2}}}} dx = \int_{t_1}^{t_2} dt \quad (2.7)$$

where,

$x(t_1)$  and  $x(t_2)$  are the particle co-ordinates at arbitrary times  $t_1$  and  $t_2$ .

Equation 2.7 is integrated to give

$$\ln \frac{A_x \left( x(t_2) - x_{i-\frac{1}{2}} \right) + v_{x_{i-\frac{1}{2}}}}{A_x \left( x(t_1) - x_{i-\frac{1}{2}} \right) + v_{x_{i-\frac{1}{2}}}} = A_x \Delta t \quad (2.8)$$

where,

$x(t_1)$  and  $x(t_2)$  are the particle x co-ordinates at time  $t_1$  and  $t_2$  and,

$\Delta t = (t_2 - t_1)$ .

Noting that from Equation 2.4

$$V_x(t_1) = A_x \left( x(t_1) - x_{i-\frac{1}{2}} \right) + V_{x_{i-\frac{1}{2}}} \quad (2.9)$$

then Equation 2.8 can be re-arranged to give

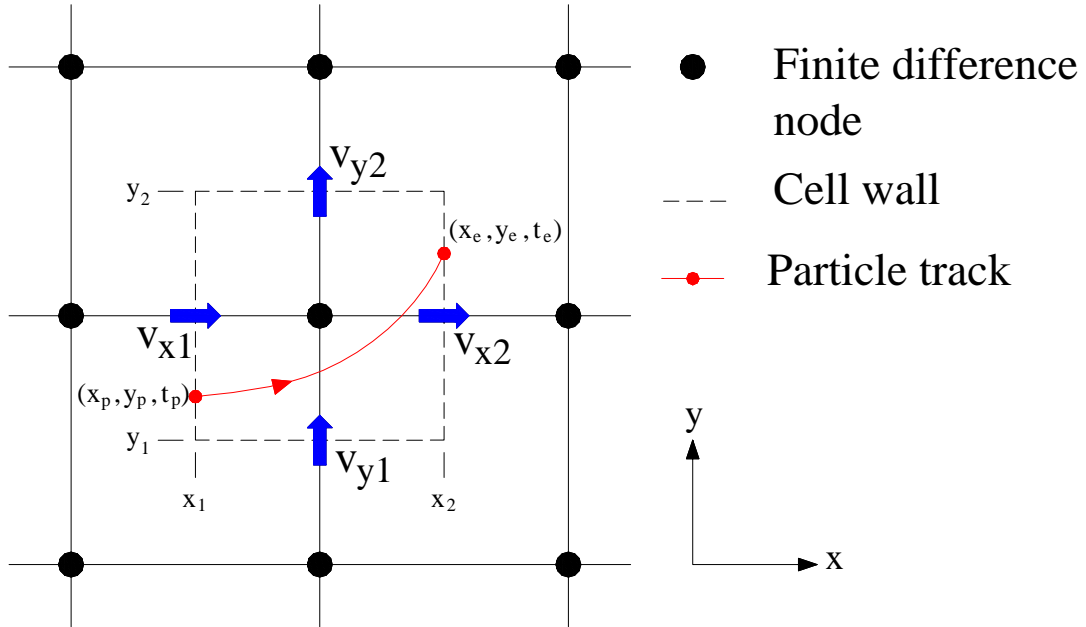
$$x(t_2) = x_{i-\frac{1}{2}} + \frac{1}{A_x} \left[ V_x(t_1) \cdot \exp(A_x \Delta t) - v_{x_{i-\frac{1}{2}}} \right] \quad (2.10a)$$

Equivalent equations to Equation 2.10a can be derived in the y and z-directions. These are

$$y(t_2) = y_{j-\frac{1}{2}} + \frac{1}{A_y} \left[ v_y(t_1) \cdot \exp(A_y \Delta t) - v_{y_{j-\frac{1}{2}}} \right] \quad (2.10b)$$

$$z(t_2) = z_{k-\frac{1}{2}} + \frac{1}{A_z} \left[ v_z(t_1) \cdot \exp(A_z \Delta t) - v_{z_{k-\frac{1}{2}}} \right] \quad (2.10c)$$

Equations 2.10a to 2.10c are used to delineate the pathline of the particle as it moves through the model domain. However, these equations are only applicable when the linear interpolation coefficients,  $A_x$ ,  $A_y$  and  $A_z$  are constant. Consequently, a particle cannot be allowed to cross a cell wall between the time  $t_1$  and time  $t_2$ . Pollock (1988) presents an efficient algorithm, which eliminates the possibility of this event by calculating the length of the tracking step that is required for a particle to travel from its current location to the cell wall through which it exits the node. For example, consider Figure 2, which shows the path of a particle in a two-dimensional model grid from its initial position  $(x_p, y_p)$  at time  $t_p$ , to the point at which it exits the cell,  $(x_e, y_e)$ , at time  $t_e$ . In this example the cell wall velocities are denoted by  $V_{x1}$  and  $V_{x2}$  in the x-direction and  $V_{y1}$  and  $V_{y2}$  in the y-direction for simplicity.



**Figure 2 Particle track through a two-dimensional cell**

To illustrate Pollock's algorithm, the assumption is made that all the cell wall velocities are greater than zero. Then if we also assume, in a first instance, that the particle leaves the cell through the wall at  $x_2$ , that is, in the positive x-direction, then Equation 2.10a can be used to calculate the length of time,  $\Delta t_x$ , that the particle takes to travel from  $x_p$  to  $x_2$ . Equation 2.10a gives

$$x_2 = x_1 + \frac{1}{A_x} \left[ v_{xp} \cdot \exp(A_x \Delta t_x) - v_{x1} \right] \quad (2.11)$$

since,

$$x(t_2) = x_2 \text{ and,}$$

$v_{xp}$  is the x-component of the velocity at the point  $(x_p, y_p)$

Rearranging Equation 2.11 gives

$$A_x(x_2 - x_1) + v_{x1} = v_{xp} \exp(A_x \Delta t_x) \quad (2.12)$$

From Equation 2.4

$$A_x(x_2 - x_1) + v_{x1} = v_{x2} \quad (2.13)$$

and therefore by substituting this in Equation 2.12 we obtain

$$\Delta t_x = \frac{1}{A_x} \ln \left( \frac{v_{x2}}{v_{xp}} \right) \quad (2.14)$$

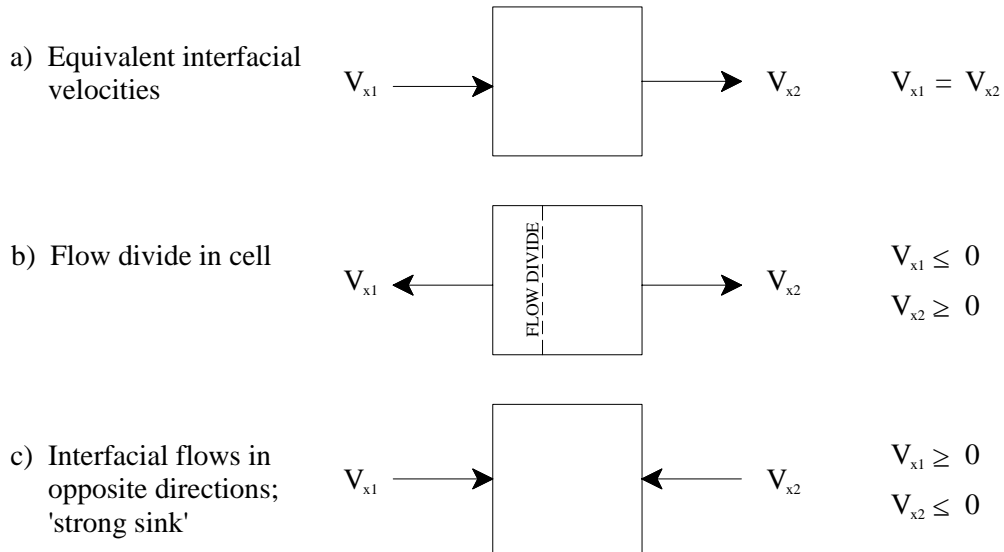
If it is assumed that the particle leaves the cell through the wall at  $y_2$ , then by a similar process the length of time,  $\Delta t_y$ , that the particle takes to travel from  $y_p$  to  $y_2$  can be derived

$$\Delta t_y = \frac{1}{A_y} \ln \left( \frac{v_{y2}}{v_{yp}} \right) \quad (2.15)$$

The comparison of  $\Delta t_x$  and  $\Delta t_y$  defines through which cell wall the particle exits. The time for the particle to exit the cell,  $\Delta t_e$ , is taken as the smaller of  $\Delta t_x$  and  $\Delta t_y$ . If  $\Delta t_x$  is smaller than  $\Delta t_y$  the particle will exit the cell through the interface at  $x = x_2$  and vice versa. If  $\Delta t_x = \Delta t_y$  then the particle will exit at the corner of the cell through the point  $(x_2, y_2)$ .

This method of moving the particle from cell wall to cell wall is easy to implement and computationally efficient. If the particle track needs to be defined in greater detail, Equations 2.10a to 2.10c can be used to define intermediate points along the pathline within the cell. This is achieved by dividing  $\Delta t_e$  by the number of intermediate steps within the cell and then using multiples of this smaller time-step to calculate the intermediate points using Equations 2.10a to 2.10c.

In the above discussion it has been assumed that the interfacial velocities are all greater than zero for simplicity. However, there are three other possible flow conditions that must be identified before the semi-analytical solution algorithm can be applied. These are shown in Figure 3.



**Figure 3 Cell wall velocity conditions needing consideration during particle tracking**

In Figure 3a, the interfacial velocities are the same. In this case Equation 2.14 is undefined and the time of travel to exit the cell must be estimated by

$$\Delta t_x = (x_2 - x_p)/v_{x1} \quad (\text{if } v_{x1} > 0) \quad (2.16a)$$

$$\Delta t_x = (x_1 - x_p)/v_{x1} \quad (\text{if } v_{x1} < 0) \quad (2.16b)$$

In Figure 3b a local groundwater divide exists. In this case, if the particle is to the left of the divide it exits to the left otherwise it exits the cell to the right. Finally, Figure 3c shows the situation in which flow is toward the cell centre from both x-directions. In this case the particle cannot leave the cell in the positive or negative x-direction. If this is simultaneously true of the flows in the y and z-directions then the node is termed a ‘strong sink’ and the particle is terminated at the cell.

### 2.3.2 Numerical integration techniques

Equation 2.2 can also be solved numerically. In fact, if a higher order interpolation scheme is used then it can only be solved using numerical methods. Numerical integration methods involve the movement of the particles in discrete *tracking steps* along the pathline. Of the numerical integration techniques, Euler’s method is the simplest. In this method the velocity at the current point is extrapolated over the tracking step. The particle is moved along the pathline using the equations

$$x' = x_p + v_{xp} \Delta t \quad (2.17a)$$

$$y' = y_p + v_{yp} \Delta t \quad (2.17b)$$

$$z' = z_p + v_{zp} \Delta t \quad (2.17c)$$

where,

$x'$ ,  $y'$  and  $z'$  are the co-ordinates of the particle’s new location,

$x_p$ ,  $y_p$  and  $z_p$  are the co-ordinates of the particle’s current location,

$v_{xp}$ ,  $v_{yp}$  and  $v_{zp}$  are the components of the particles velocity at its current location and,

$\Delta t$  is the length of the tracking step.

Whilst Euler’s method is straightforward to implement, the length of the tracking step,  $\Delta t$ , must generally be small to maintain accuracy. This is because the velocity is extrapolated over the tracking interval. A numerical method with a higher order of accuracy is that of the Runge-Kutta technique. This is implemented in ZOOPT in addition to the semi-analytical technique. The Runge-Kutta method moves a particle over a tracking step,  $\Delta t$ , by combining information from a number of Euler-type steps. It is generally more accurate than Euler’s method but computationally less efficient, however, the Runge-Kutta method need only be implemented in ZOOPT in a few specific situations. These are discussed in detail in Section 3 of this report.

In the Runge-Kutta method the velocity is calculated four times for each tracking step: once at the current particle location, twice at two trial midpoints and once at a trial end point. With reference to Figure 4, a two-dimensional case, the process is defined using the following equations

$$x_{n+1} = x_n + \frac{(v_{xp1} + 2v_{xp2} + 2v_{xp3} + v_{xp4})}{6} \Delta t \quad (2.18a)$$

$$y_{n+1} = y_n + \frac{(v_{yp1} + 2v_{yp2} + 2v_{yp3} + v_{yp4})}{6} \Delta t \quad (2.18b)$$

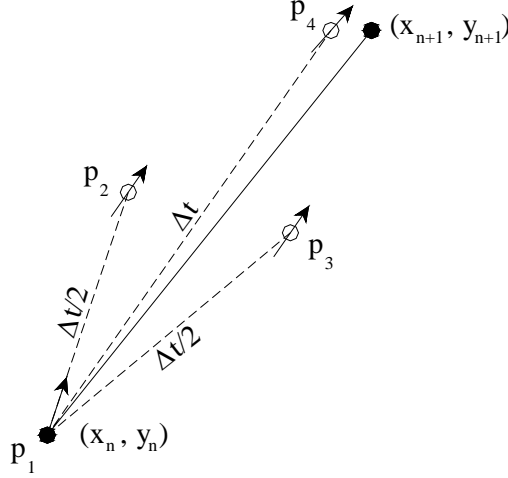
where,

$(v_{xpi}, v_{ypi})$  are the velocity components at the points  $(x_{pi}, y_{pi})$  for  $i = 1$  to 4,

$\Delta t$  is the length of the tracking step and,

$(x_{n+1}, y_{n+1})$  is the final position of the particle at the end of the tracking step.

Equations 2.18a and 2.18b cannot be applied directly because the velocities,  $v_{xpi+1}$  and  $v_{ypi+1}$  depend on the co-ordinates of  $x_{pi}$  and  $y_{pi}$ . Hence, the co-ordinates of the particle at the end of the tracking step are calculated iteratively.



**Figure 4 Intermediate steps taken during fourth-order Runge-Kutta method**

The velocities at each of the points  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$  are calculated using linear interpolation of the cell wall velocities after the co-ordinate of each previous trial point has been calculated. The co-ordinates of these points are calculated by performing the following steps. The co-ordinates of  $p_2$  are calculated using

$$x_{p2} = x_{p1} + v_{xp1} \Delta t / 2 \quad (2.19a)$$

$$y_{p2} = y_{p1} + v_{yp1} \Delta t / 2 \quad (2.19b)$$

from which the co-ordinates of the point  $p_3$  are subsequently calculated using

$$x_{p3} = x_{p1} + v_{xp2} \Delta t / 2 \quad (2.20a)$$

$$y_{p3} = y_{p1} + v_{yp2} \Delta t / 2 \quad (2.20b)$$

and then finally the co-ordinates of the point  $p_4$  are determined using

$$x_{p4} = x_{p1} + v_{xp3} \Delta t \quad (2.21a)$$

$$y_{p4} = y_{p1} + v_{yp3} \Delta t \quad (2.21b)$$

By repeating this procedure the particle is moved through the model domain until it reaches a discharge point. The above equations are easily extended for application of the technique to three-dimensional problems as in ZOOPT, the particle tracking code developed here.



Whilst the Runge-Kutta technique incorporates a higher order of accuracy than the simpler Euler's method, it is computationally less efficient. Consequently, it is important to optimise the length of the tracking step during the procedure to both maintain accuracy and minimise computational effort. This is performed in ZOOPT using the 'step doubling' procedure presented by Zheng and Bennett (1995). In this procedure the tracking step is performed twice. First the tracking step is made using a time interval of  $\Delta t$  and then it is repeated by taking two steps of half the length i.e.  $\Delta t / 2$ . The distance,  $\Delta s$ , between the two points calculated using a full step and two half steps is used to adjust the full length of the tracking step. As Zheng and Bennett (1995) state, if the fourth-order Runge-Kutta technique is used, the tracking solution is accurate to the fourth-order, hence,  $\Delta t$  can be scaled as  $(\Delta s)^{\frac{1}{5}}$ . Equation 2.22 is used to calculate the required tracking step size,  $\Delta t_0$ , that will yield an error less than  $\Delta s_0$ , given an initial calculation of  $\Delta s$  using an initial tracking step of  $\Delta t$ . The term  $f_s$  is a safety factor and is given a value slightly smaller than one e.g. 0.9.

$$\Delta t_0 = f_s \Delta t \left( \frac{\Delta s_0}{\Delta s} \right)^{\frac{1}{5}} \quad (2.22)$$

### 3 Functionality of the particle tracking code, ZOOPT

#### 3.1 INTRODUCTION

The particle tracking code ZOOPT has been developed for use in conjunction with the object-oriented regional groundwater model ZOOMQ3D. ZOOMQ3D incorporates the mechanisms that are required to satisfy generally-accepted functional requirements of a commonly-applied regional groundwater flow model. The model is described in detail in Jackson (2001) and Jackson (2002). In addition to representing hydrogeological features, such as rivers and pumped wells, that are commonly simulated using groundwater flow models, ZOOMQ3D incorporates the vertical variation of hydraulic conductivity with depth (Taylor et al., 2001) and local grid refinement. Particle tracking within ZOOPT is compatible with all of these mechanisms.

At this stage of development, ZOOPT enables steady-state particle tracking under advective transport in both the forward and reverse directions. Consequently, the code can be applied to the definition of borehole catchments and associated source protection zones, the identification of recharge and discharge areas and the visualisation of groundwater flow patterns, for example. Back tracking is easy to implement within a particle tracking code. In ZOOPT the inter-nodal flows are simply multiply by minus one as they are read from the node-by-node flow input file. In addition to tracking particles using steady-state simulations, the code can be used to visualise unsteady flow fields at a particular instants in time. Steady-state particle tracking can be applied easily at any time step of a time-variant simulation. The flows calculated at the end of a time step are used to define an approximate borehole catchment by implementing steady-state particle tracking routine. This approach can be useful to analyse, for example, the approximate change in shape of a borehole catchment during a seasonal recharge or abstraction cycle.

Particles are tracked using the semi-analytical technique described above. However, around particular model features, for example 'weak' sink nodes or nodes which exhibit a vertical variation of hydraulic conductivity, particles have to be tracked using the Runge-Kutta method. The switch between the use of the semi-analytical and Runge-Kutta methods is made automatically within the model code. Though the Runge-Kutta technique is only implemented occasionally, the user can enforce its continuous use. This option provides an alternative to the

semi-analytical technique, though, the semi-analytical method is computationally more efficient and should be used in preference to Runge-Kutta tracking.

ZOOPT is designed to track particles in models where the horizontal hydraulic conductivity varies with depth (VKD) and this is one feature that requires the application of the Runge-Kutta method. In this case the horizontal velocity depends on the hydraulic conductivity at the elevation of the particle within the node. Consequently, the integral in Equation 2.2 cannot be determined analytically and a fully numerical tracking method must be employed.

In addition to the application of the particle tracking code to VKD nodes, the code is compatible with the local grid refinement technique incorporated within ZOOMQ3D. Local grid refinement enables the zooming of the mesh within discrete areas of a model grid to increase accuracy or model detail. ZOOPT tracks particles through these locally refined grids.

The occurrence of ‘weak’ sinks, which is a problem associated with particle tracking is circumvented by ZOOPT. Weak sinks are commonly generated when, for example, an abstraction well, which distributes its effect over the volume of the cell, is not sufficiently strong to cause groundwater to flow into the associated finite difference node through all of its faces. In this case, it is not possible to determine whether a particle should leave the cell through one of its walls or whether the particle should terminate at the well. An approach is adopted in ZOOPT that eliminates the problems associated with weak sinks. This is discussed in Section 3.2.

Whilst ZOOPT currently incorporates significant functionality, it can be developed further. Particles cannot be tracked for a specific length of time, for example, to define a fifty-day capture zone for an abstraction well. However, this would be relatively simple to develop. The ability to track particles time-variantly is important but ZOOPT does not currently have this capability. Time-variant particle tracking is generally performed by repeating a number of steady-state tracking steps during which particle velocity is assumed to be constant over time. This approach is adopted in MODPATH. However, Lu (1994) presents a semi-analytical approach in which velocity is linearly interpolated over both space and time. This is a more accurate technique and should be relatively straightforward to develop in ZOOPT.

## **3.2 VELOCITY CALCULATION**

### **3.2.1 Weak sinks**

A problem that can be associated with particle tracking codes is that of ‘weak’ sinks. In finite difference models groundwater can flow out of a node through a sink that is distributed throughout the whole volume of the cell. For example, abstraction wells, rivers, drains and springs are all simulated in ZOOMQ3D using such sinks. If the discharge rate of the sink is sufficient, water will flow into the node across all of its six faces. This is termed a ‘strong’ sink. However, if the discharge rate of the sink is insufficient to cause inflow across all sides of the node, that is, water flows out of the cell across one or more of its six faces, then a weak sink is generated. Weak sinks present a problem because it is not possible to determine whether a particle leaves the node through a cell wall or through the sink. This is because the calculation of the velocity within the cell is based on the discharge rates across the cell interfaces only and does not take account of the effect of a distributed sink on the velocity field.

Weak sinks caused by abstraction wells are dealt with separately in the next section. With regard to the other model features listed above that can cause weak sinks, the problem can be circumvented by assuming that the discharge to the sink actually occurs through one of the cell walls. In effect, therefore, the sink is removed from the node and one of the cell wall velocities is re-calculated. In ZOOPT the flows between the aquifer and rivers, head dependent leakage nodes (‘drains’) and springs, which are represented as distributed sinks, are all assigned to the upper face of the corresponding finite difference node. The velocity is then recalculated across the node’s upper face.

Recharge is dealt with in a similar manner. Recharge is assumed to fall vertically onto the aquifer. Consequently, the velocity across the upper face of the node is adjusted to account for this inflow. Abstraction wells are the other features of ZOOMQ3D that can generate weak sinks. These are less straightforward to deal with and are thus discussed separately in the next section.

### 3.2.2 Weak sinks caused by abstraction wells

As stated in the last section, the creation of weak sinks by mechanisms other than wells is dealt with by assigning the discharge to one of the walls of the finite difference node. The selection of the appropriate wall to which the flow is assigned is based on physically justifiable assumptions, for example, groundwater recharge arrives at the water table from above. However, with regard to abstraction wells, such an assumption is not justifiable because groundwater is drawn towards wells from all directions. Consequently, abstraction wells present a more significant problem. To identify if a particle terminates at a weak sink well Zheng (1994) uses a special velocity interpolation scheme within the corresponding finite difference node. This is based on the superposition of an analytical solution for radial flow to a well and a solution for unidirectional regional groundwater flow. The interpolation scheme alters the velocity components in the x and y-directions but continues to use linear interpolation in the z-direction. With reference to Figure 5, the horizontal components of the particles velocity are given by

$$V_{xp} = \frac{Q'_w \sqrt{a}}{2\pi(z_2 - z_1)\theta} \left[ \frac{x_p}{x_p^2/a + y_p^2} \right] + \frac{(V_{x1} + V_{x2})}{2} \quad (3.1)$$

$$V_{yp} = \frac{Q'_w \sqrt{a}}{2\pi(z_2 - z_1)\theta} \left[ \frac{y_p}{x_p^2/a + y_p^2} \right] + \frac{(V_{y1} + V_{y2})}{2} \quad (3.2)$$

where

$x_p$  and  $y_p$  are the particle's x and y co-ordinates with respect to the centre of the node,

$V_{xp}$  and  $V_{yp}$  are the x and y components of the particle's velocity,

$V_{x1}$ ,  $V_{x2}$ ,  $V_{y1}$ ,  $V_{y2}$ ,  $V_{z1}$ , and  $V_{z2}$  are the x, y and z components of the cell wall velocities,

$$Q'_w = Q_{\text{WELL}} - [(V_{z1} - V_{z2}) \cdot (x_2 - x_1) \cdot (y_2 - y_1)],$$

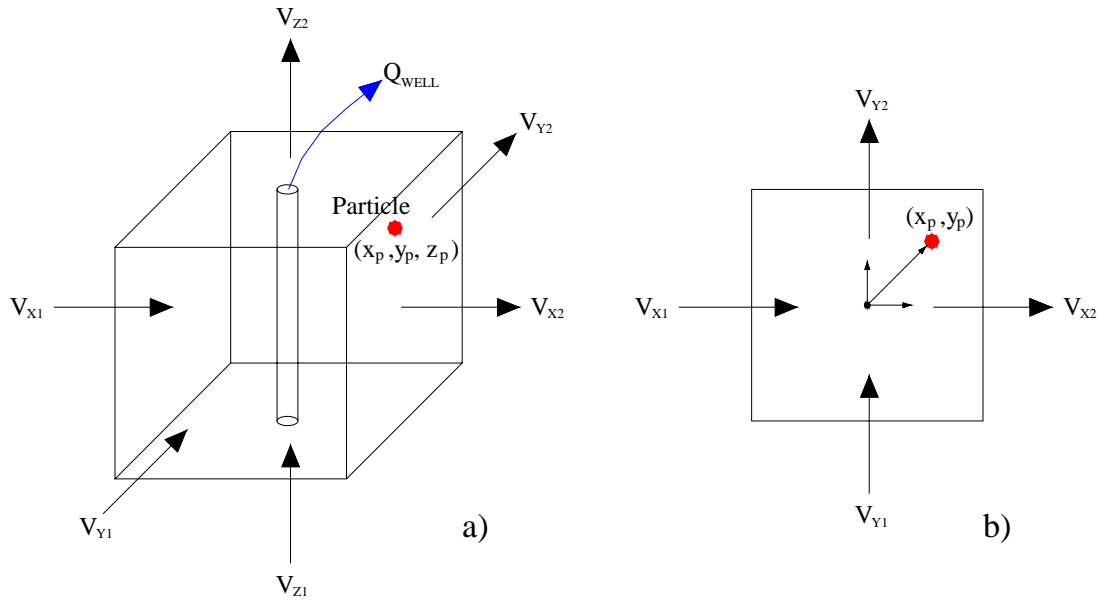
$Q_{\text{WELL}}$  is the abstraction rate of the well,

$x_1$ ,  $x_2$ ,  $y_1$ ,  $y_2$ ,  $z_1$  and  $z_2$ , are the x, y and z co-ordinates of the cell walls,

$a = (K_y/K_x)$  is the ratio of hydraulic conductivities in the x and y-directions and,

$\theta$  is the porosity of the node.

This interpolation scheme forces particles either to converge towards the well or to leave the finite difference node through one of its interfaces. Particles are terminated if they enter within the radius of the well. The scheme is implemented in ZOOPT and is an elegant solution to the problem of weak sink wells. At wells that generate strong sinks this interpolation scheme is not applied and all particles are terminated as they enter the node.



**Figure 5 Illustration of a well as a distributed sink at the centre of a finite difference node showing surrounding cell walls in a) three dimensions and b) plan view**

### 3.3 GRID CONSIDERATIONS

#### 3.3.1 Distorted vertical discretisation

Because of efficiency considerations finite difference models are often constructed in which the elevation of the top and bottom of model layers varies over the model domain. This variation generally results from the need to approximate the changing shape of hydrogeological units. However, vertical distortion of the mesh causes problems because the particle tracking solution is based on a fixed orthogonal grid. For example, consider that it is calculated that a particle leaves a model cell in the horizontal and positive x-direction. If the particle leaves towards the top of the cell and the grid is distorted then it is possible for the particle to enter a node contained in an upper layer. This situation is shown in Figure 6a. If this occurs the particle position must be corrected before the next particle move is made.

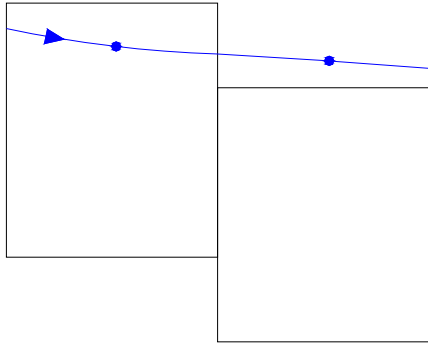
When the semi-analytical solution is adopted the particles elevation is adjusted on the interface between nodes, that is, when it passes from one node to the next. At this point the local  $z$  co-ordinate of the particle prior to the correction, with respect to the top and bottom elevations of the first node, must be equal to its local  $z$  co-ordinate in the node it is entering after the correction. This calculation is shown in Figure 6b. Because of this correction the particle path can appear unsmooth when plotted. This problem is inherent in the representation of three-dimensional models as vertically varying layers.

When the Runge-Kutta method is used the particles position must be modified after the tracking step within the node the particle has entered. The required vertical correction is calculated by considering that if the vertical velocity component was zero, the particle's local  $z$  co-ordinate within each cell would have remained the same. The correction factor,  $\Delta z_c$ , is calculated using this assumption and is given by

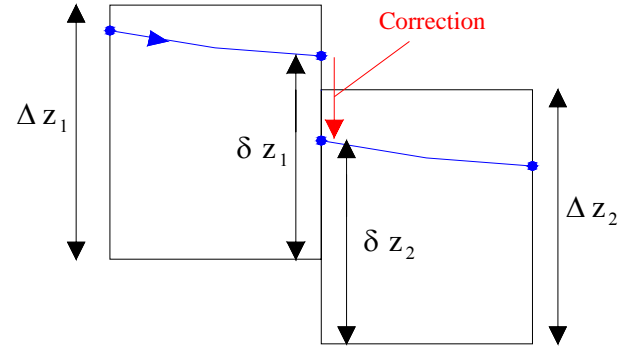
$$\Delta z_c = \frac{\Delta z_2}{\Delta z_1} (z_n - z_1) + z_2 - z_n \quad (3.3)$$

The correction procedure is illustrated in Figure 6c. A full derivation of this correction term is presented by Zheng (1994).

a) Uncorrected particle path

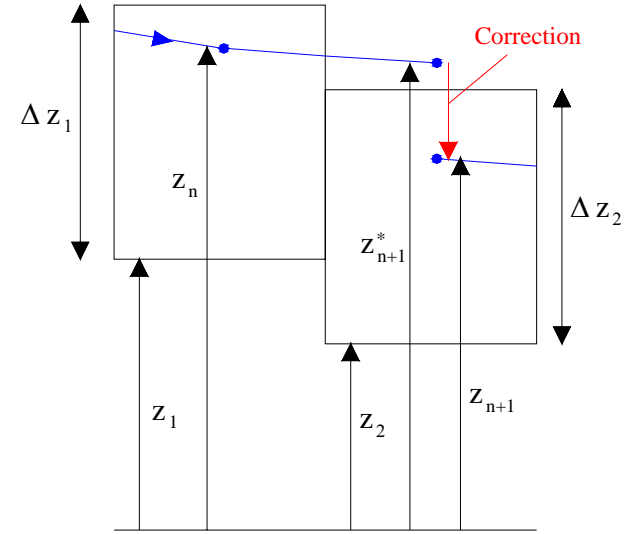


b) Semi-analytical solution  
Correction at nodal interface



$$\frac{\delta z_1}{\Delta z_1} = \frac{\delta z_2}{\Delta z_2}$$

c) Runge-Kutta solution  
Correction after tracking step



$$z_{n+1} = z_{n+1}^* + \frac{\Delta z_2}{\Delta z_1} (z_n - z_1) + z_2 - z_n$$

**Figure 6 Correction of particle elevation in vertical distorted model layers**

### 3.3.2 Unconfined aquifer layers

In phreatic aquifer layers the elevation of the water table determines the vertical thickness of the finite difference node. The elevation of the water table is defined as the simulated groundwater head. Consequently, in unconfined model layers the nodes are distorted vertically again. The position of particles that are tracked through unconfined nodes are corrected in the same way as described for fixed but vertically distorted grid nodes.

### 3.4 QUASI THREE-DIMENSIONAL LAYERING

Not all hydrogeological layers are always included explicitly in a groundwater model. Consider that a groundwater system is composed of a sequence of high permeability horizontal layers that are separated by a series of low permeability aquitards. Often these aquitards are not represented by a series of finite difference nodes in a groundwater model. This is because the assumption can be made that the flow in the low permeability layers will be essentially vertical. Consequently, low permeability layers are commonly modelled by adjusting the vertical conductance between the two adjacent aquifer layers. When this approach is adopted the model is stated to be *quasi three-dimensional*.

Whilst this method of representing aquitards in groundwater models is computationally efficient, the particle tracking routine has to be modified because there are no grid nodes associated with these low permeability layers. ZOOPT recognises the presence of quasi three-dimensional layers and moves the particles vertically through them. The time of travel through the low permeability layer is calculated from the leakage between the two adjacent simulated aquifers, however, the user must specify the porosity of each aquitard.

### 3.5 VERTICAL VARIATION OF HYDRAULIC CONDUCTIVITY WITH DEPTH

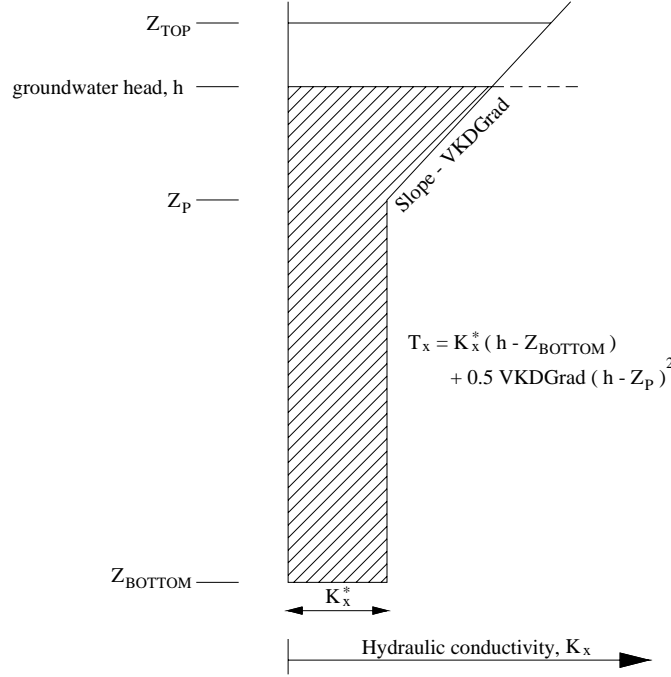
The flow model ZOOMQ3D incorporates the variation of hydraulic conductivity with depth within layers of the finite difference grid. This representation of the vertical variation of hydraulic conductivity provides an alternative to the development of multi-layer models, in which individual layers are characterised by uniform horizontal hydraulic conductivity in the vertical direction.

The approach has been developed to enable the more accurate description of the variation of hydraulic conductivity in limestone, and particularly Chalk aquifers, in which higher hydraulic conductivity values are often associated with the zone of fluctuation of the water table. The method circumvents numerical difficulties that are related to the de-watering of layers in multi-layer models. The variation of the horizontal hydraulic conductivity with depth is defined by profiles such as that shown in Figure 7.

A VKD profile describes the change in hydraulic conductivity with depth at a particular point in the aquifer. Profiles are defined by two sections. In the lower section, between  $Z_{\text{BOTTOM}}$  and  $Z_p$  in Figure 7, hydraulic conductivity is constant. In the upper section, between  $Z_p$  and  $Z_{\text{TOP}}$ , hydraulic conductivity increases linearly with elevation. Because different values of hydraulic conductivity can be specified in the two orthogonal horizontal directions (x and y), six values are used to parameterise the profile:

- i. The elevation of the base of the profile,  $Z_{\text{BOTTOM}}$ .
- ii. The elevation of the top of the profile,  $Z_{\text{TOP}}$ .
- iii. The elevation of the point of inflection,  $Z_p$ .
- iv. The hydraulic conductivity in the x direction,  $K_x^*$ , below  $Z_p$ .

- v. The hydraulic conductivity in the y direction,  $K_y^*$ , below  $Z_p$ .
- vi. The gradient of the profile above  $Z_p$ , VKDGrad. This is equal to the increase in hydraulic conductivity per metre rise in elevation.



**Figure 7 Parameters used to define VKD profiles in ZOOMQ3D**

The value of the VKDGrad parameter may be either negative, zero or positive. Consequently, in addition to an increase in hydraulic conductivity with depth above  $Z_p$ , hydraulic conductivity can be specified to decrease or remain constant. To calculate transmissivity the following equations are used:

$$T_x = K_x^* (h - Z_{\text{BOTTOM}}) + 0.5 \text{VKDGrad} \cdot (h - Z_p)^2$$

$$T_y = K_y^* (h - Z_{\text{BOTTOM}}) + 0.5 \text{VKDGrad} \cdot (h - Z_p)^2$$

for  $h > Z_p$ , and,

$$T_x = K_x^* (h - Z_{\text{BOTTOM}})$$

$$T_y = K_y^* (h - Z_{\text{BOTTOM}})$$

for  $h \leq Z_p$ , where  $h$  is the water table elevation.

At those nodes of the finite difference grid where hydraulic conductivity varies with elevation the Runge-Kutta particle tracking technique must be used to define pathlines. The integral in Equation 2.2 cannot be evaluated analytically because the horizontal velocity varies in the  $z$ -direction, that is, towards the top of the VKD profile the horizontal velocity is greater than towards its base. Consequently the semi-analytical method cannot be used. Within ZOOPT the assumption is made that the horizontal velocity of the particle is proportional to the horizontal

hydraulic conductivity at its location. With reference to Figure 8, the component of the velocity in the x-direction at the cell walls at  $x_1$  and  $x_2$  are given by

$$V_{x1}(z) = \frac{K_x(z)}{\bar{K}_x} \cdot \frac{Q_{x1}}{\theta \Delta y (z_2 - z_1)} \quad (3.4)$$

$$V_{x2}(z) = \frac{K_x(z)}{\bar{K}_x} \cdot \frac{Q_{x2}}{\theta \Delta y (z_2 - z_1)} \quad (3.5)$$

where,

$V_{x1}(z)$  and  $V_{x2}(z)$  are x-component of velocity (m/day) on the cell walls at elevation  $z$ ,

$Q_{x1}$  and  $Q_{x2}$  are flow rates entering and exiting the cell in the x-direction ( $\text{m}^3/\text{day}$ ),

$K_x(z)$  is the hydraulic conductivity (m/day) in the x-direction at elevation  $z$ ,

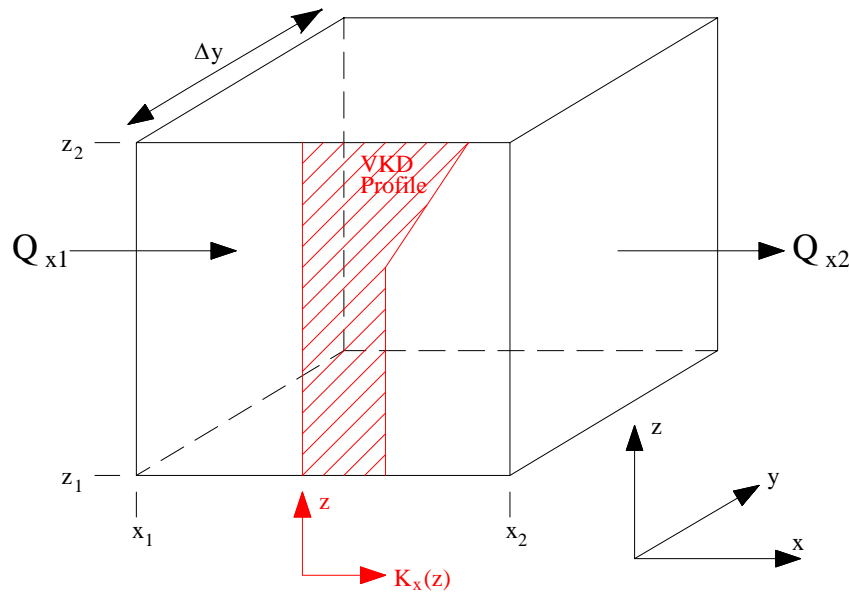
$\bar{K}_x = \frac{1}{(z_2 - z_1)} \int_{z_1}^{z_2} K_x(z) dz$ , the mean hydraulic conductivity (m/day) in the x-direction,

$\theta$  is the porosity of the node,

$\Delta y$  is the width of the node in the y-direction (m) and,

$z_1$  and  $z_2$  are the elevations of the bottom and top of the node (m).

Similar equations are written for the component of velocity in the y-direction. However, because the hydraulic conductivity in the z-direction is considered uniform throughout the node, no modification is made to way in which the z-component of velocity is calculated. ZOOPT recognises when a particle enters a node in which hydraulic conductivity varies with depth and then invokes the use of the Runge-Kutta technique and the application of Equations 3.4 and 3.5.



**Figure 8 Illustration of the interpolation of velocity in VKD nodes**



## 4 Model input and output

### 4.1 MODEL INPUT

Most of the input files that are required by ZOOPT are also used by the flow model ZOOMQ3D. The particle tracking code requires those ZOOMQ3D input files that describe the structure of the model and the aquifer's hydraulic parameters. In addition to this, two of ZOOMQ3D's output files, 'flowbal.txt' and 'heads.txt', are used as input for ZOOPT. Some additional input files are required by ZOOPT alone. The names and descriptions the input files that are specific to ZOOPT are described below.

**Table 2 ZOOPT input files**

File name	File description
<b>zoopt.dat</b>	// Forward (f) or back tracking (b) f
The main ZOOPT input file	// Number of intermediate tracking points within cell (>=0) 0
	// Runge-Kutta safety factor and error criterion 0.9 0.00001
	// Enforce use of Runge-Kutta (y/n) n
	// DXF drawing z scale factor (>=1.0) 100.0
	// Draw grid in DXF file (y or n) y
	// Time step, month & year for which to track particles 3 12 4
<b>particles.dat</b>	50
Particle definition file	1000.0      1000.0      90.0 0
	1000.0      1100.0      90.0 0
	1000.0      1200.0      90.0 0
	etc ... in format
	x      y      z      layer number
	If layer number is zero, z is a global co-ordinate otherwise z is a local co-ordinate between zero and one within the specified layer
<b>porosity#.cod</b>	Porosity definition code file. One for each model layer i.e. porosity1.cod, porosity2.cod etc. ASCII files containing factors which are used to define porosity values within zones of the model grid
<b>porosity#.map</b>	Porosity definition map file. One for each model layer i.e. porosity1.map, porosity2.map etc. ASCII files containing a map of the model grid which is used to define different zones of porosity
<b>porosity#a.cod</b>	Porosity definition code file. One for each quasi three-dimensional layer i.e. porosity1a.cod, porosity2a.cod etc. These are aquifer layers that are simulated implicitly by adjusting the vertical conductance between adjacent layers of finite difference nodes.
<b>porosity#a.map</b>	Porosity definition map file. One for each quasi three-dimensional layer i.e. porosity1a.map, porosity2a.map etc. These are aquifer layers that are simulated implicitly by adjusting the vertical conductance between adjacent layers of finite difference nodes.
<b>flowbal.txt</b>	Node-by-node flow balance file output by ZOOMQ3D. Contains flow balance information for each node of the grid for each time-step of the simulation
<b>heads.txt</b>	Node-by-node groundwater head file output by ZOOMQ3D. Contains groundwater head information for each node of the grid for each time-step of the simulation

## 4.2 MODEL OUTPUT

ZOOPT produces six output files, two giving information on particle tracks, three DXF files used to plot the pathlines for visual examination and an error message file. These are described below.

**Table 3 ZOOPT output files**

Output file name and description	
<b>tracks.out</b>	Output pathline data in ASCII format  Particle number, x, y, z, total travel time, total distance travelled  One line for each move of each particle.
<b>ptend.out</b>	Particle end point output data in ASCII format  Particle number, (x,y,z) of end position, total time, total distance, (x,y,z) of start position  One line for each particle.
<b>tracks.dxf</b>	Three dimensional DXF file used to draw model grid and pathlines. Co-ordinates in x,y,z order. Viewed in plan view when opened in a software package that only reads DXF files in two dimensions.
<b>tracks_xz.dxf</b>	Three dimensional DXF file used to draw model grid and pathlines. Co-ordinates in x,z,y order. Viewed from front when opened in a software package that only reads DXF files in two dimensions.
<b>tracks_yz.dxf</b>	Three dimensional DXF file used to draw model grid and pathlines. Co-ordinates in y,z,x order. Viewed from side when opened in a software package that only reads DXF files in two dimensions.
<b>zoopt.err</b>	Error file giving details of any errors during particle initialisation or tracking

## 4.3 RUNNING THE MODEL

Running the particle tracking ZOOPT model is straightforward. The following procedure is undertaken:

- Run the flow model ZOOMQ3D.
- Copy the ZOOMQ3D input files and the output files “heads.txt” and “flowbal.txt” into the ZOOPT directory.
- Create the ZOOPT input files listed in Section 4.1.
- Run ZOOPT.

## 5 Model testing

### 5.1 INTRODUCTION

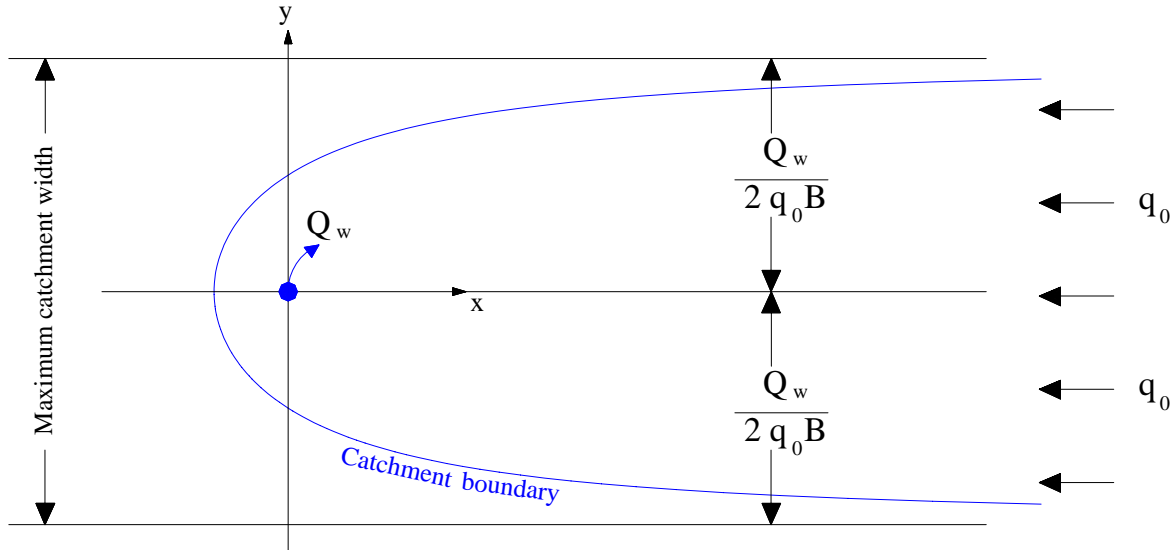
The following particle tracking simulations are designed to illustrate the operation of ZOOPT. A rigorous testing procedure has been undertaken in which numerous particle tracking tests have been performed using different flow models. The presentation of all of these simulations would be excessive and consequently a select set of tests is described next. These illustrate the application of the particle tracking code to different flow regimes and to models containing different hydrogeological mechanisms. The first example is a comparison of the model with an analytical solution to the definition of a borehole catchment.

### 5.2 TEST 1: ANALYTICAL SOLUTION

Bear (1979) presents an analytical solution for the definition of the size of a catchment around a well, which pumps at a constant rate in a uniform regional flow field. The solution considers a single abstraction well, of infinitesimally small diameter, pumping at a constant rate in a homogeneous, isotropic, confined aquifer of constant thickness. The well pumps from the aquifer in which there is a constant uniform flow in the negative x-direction, as illustrated in Figure 9. The figure shows the groundwater divide, which delineates the area of the aquifer that supplies the entire discharge of the well. The equation of this groundwater divide, or streamline is

$$y/x = \pm \tan(2\pi q_0 B y / Q_w), \quad + \text{ for } y > 0, - \text{ for } y < 0 \quad (5.1)$$

where  $q_0$  is the regional specific discharge rate (m/day),  $B$  is the aquifer thickness (m) and,  $Q_w$  is the well pumping rate ( $\text{m}^3/\text{day}$ ). The maximum width of the borehole catchment is  $Q_w/q_0 B$ , which is approached asymptotically upstream of the pumping well.

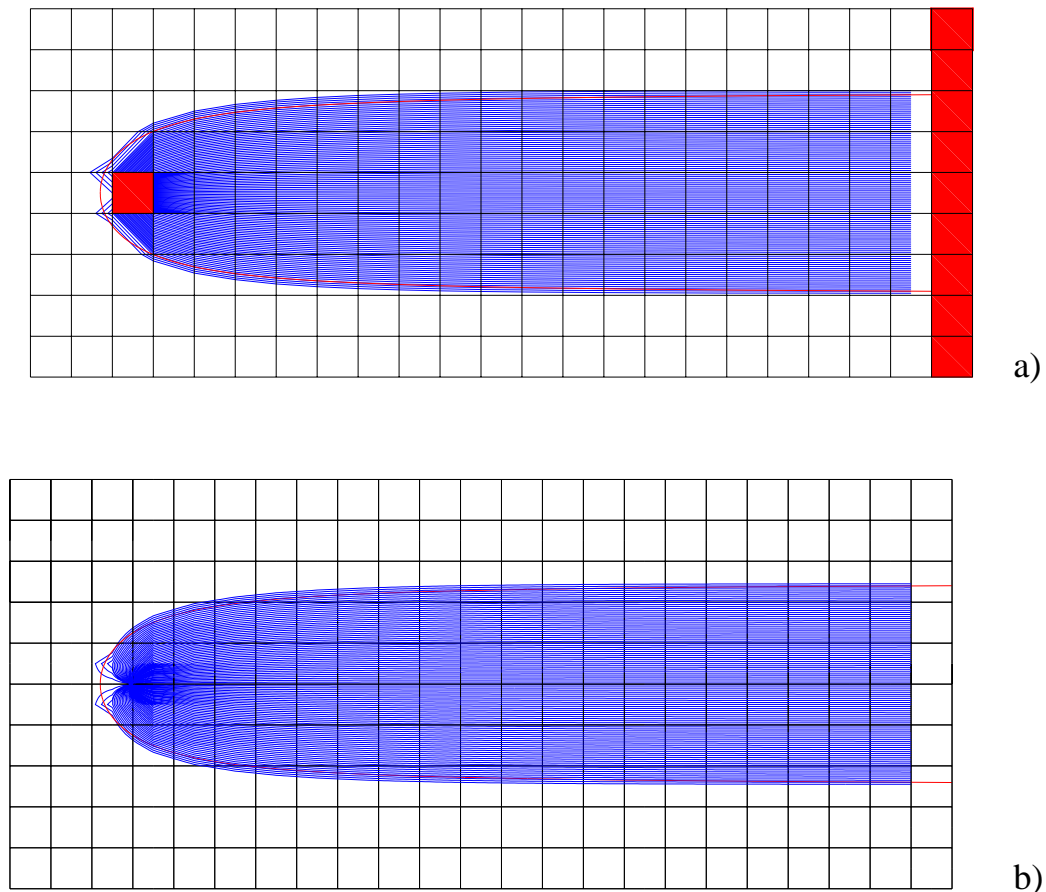


**Figure 9 A single pumping well in a uniform flow field, after Bear (1979)**

To test the accuracy of ZOOPT a model is developed to define the borehole catchment of a well in such a groundwater system. This is compared to the analytical solution and to a corresponding MODFLOW (McDonald and Harbaugh, 1988) model of the aquifer. Details of the model grids around the borehole catchment are shown in Figure 10 for both the MODFLOW/MODPATH and ZOOMQ3D/ZOOPT models. Both models are based on a uniform 500 m square mesh, though ZOOMQ3D is node-centred and MODFLOW is block-centred, which accounts for the 250 m offset between the two model grids in Figure 10 to align the pumping wells.

The uniform regional specific discharge,  $q_0$ , is 0.02 m/day, the aquifer is 100 m thick and the well pumps at a constant rate of 500 m<sup>3</sup>/day. The hydraulic conductivity of the aquifer is 10 m/day. In this case the analytical solution states that the maximum width of the catchment is 2500 m. In both models one hundred particles are defined in a uniform circle around the well of radius 50 m. These are tracked in the reverse direction to define the borehole catchment. Their pathlines are shown in blue in Figure 10. Also plotted in the figure is the analytical solution of the groundwater divide. This is shown in red.

The particles are tracked 9500 m upstream of the well. At this point, the width of the catchment should be 2400.01 m wide. The width of the catchment defined by the pathlines of the ZOOPT and MODPATH models are 2450 m and 2500 m, respectively. Both models simulate the shape of the catchment adequately as is illustrated graphically by Figure 10. However, one point to note is the slight irregular nature of the pathlines arriving at the well from the left. These are not smooth because of the relatively large scale of the model mesh. Smoother pathlines are generated if a finer mesh is used around the abstraction well, for example, see Figure 15.



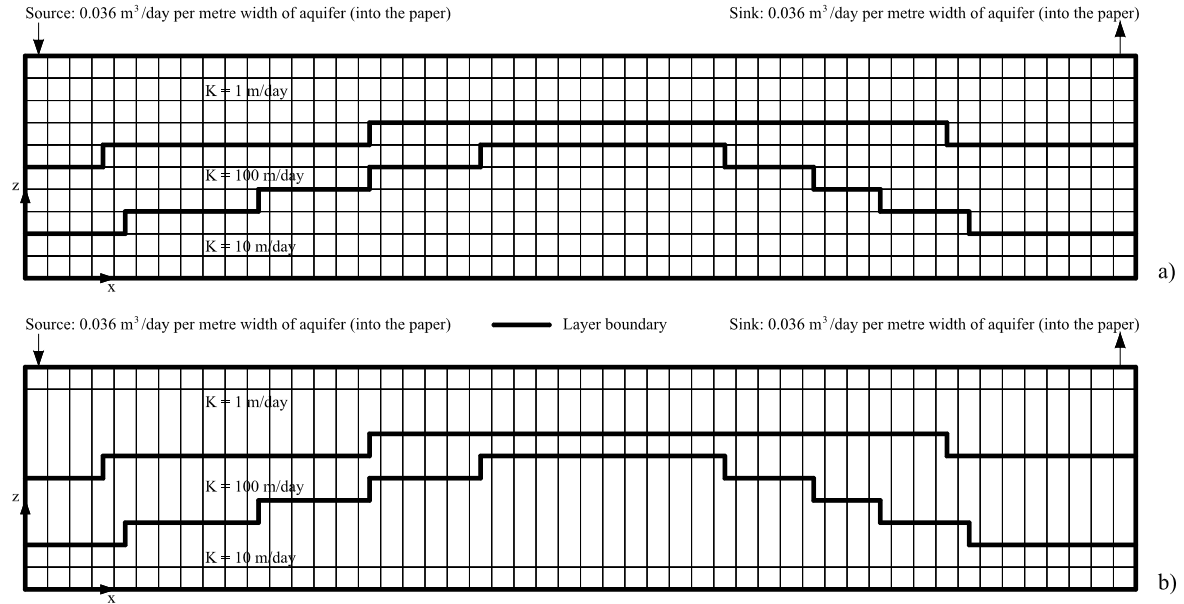
**Figure 10 Comparison of the analytical solution for the shape of a borehole catchment (red) with particle tracks of (a) the MODFLOW model and (b) the ZOOPT model**

### 5.3 TEST 2: NON-RECTANGULAR VERTICAL DISCRETISATION

To test if particles are tracked correctly through models that contain vertically distorted meshes a simulation is performed using a model similar to that presented by Zheng (1994). This test compares a model using a regular, fine mesh with one incorporating vertically distorted grids. The method of tracking of particles through such grids is discussed in Section 3.4.

The models represent a cross-section through a confined aquifer, which is 500 m long (in the x-direction) and 100 m thick (in the z-direction). The aquifer contains three hydrogeological layers which vary in thickness along its length and which have different hydraulic conductivities. The hydraulic conductivities of the top, middle and bottom layers are 1 m/day, 100 m/day and 10 m/day, respectively as shown in Figure 11.

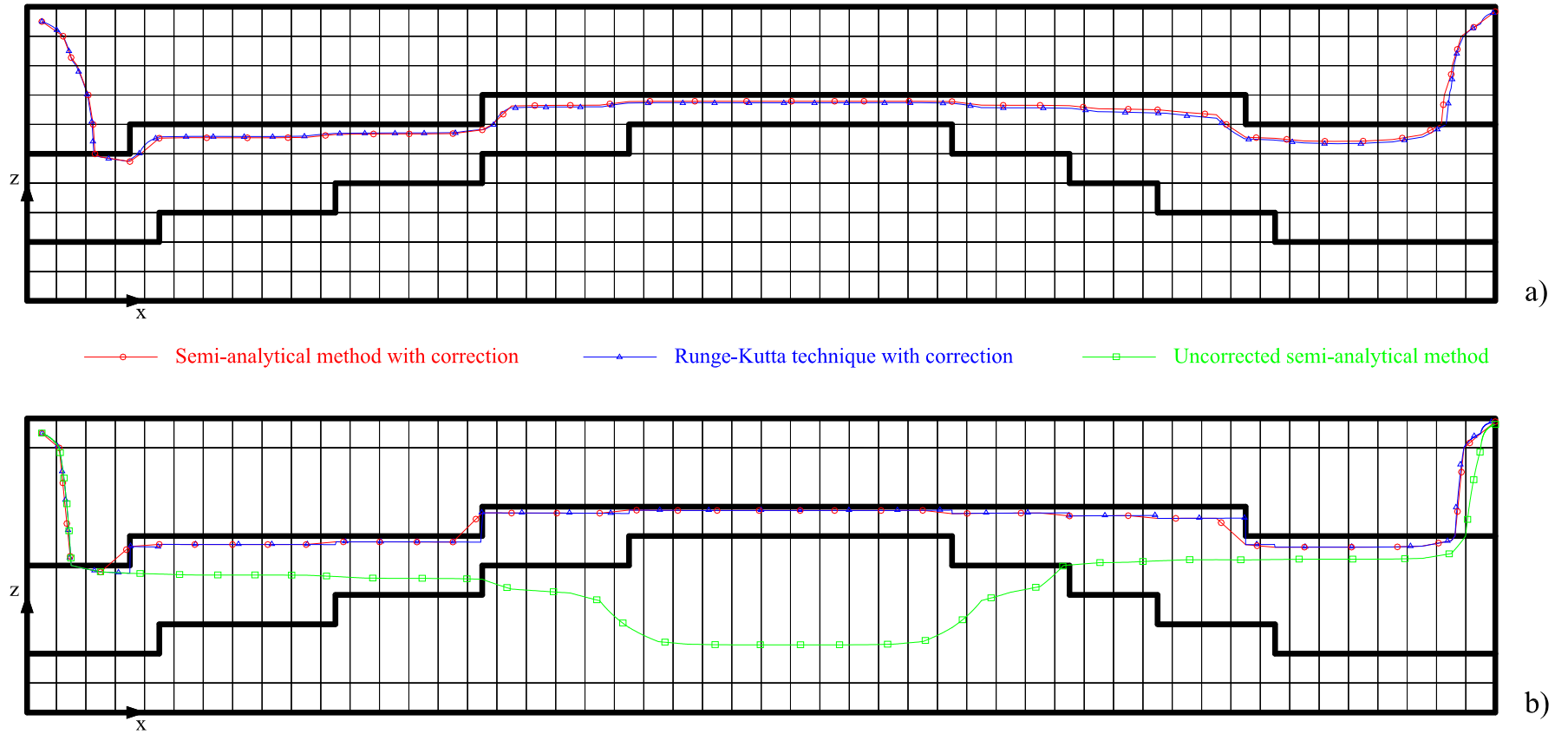
The finite difference grids for the two models are shown in Figure 11. In the first model, Figure 11a, the grid is composed of a uniform 10 m square mesh. Ten layers of finite difference nodes are used throughout the model domain, again each of which is 10 m thick. In the second model, Figure 11b, only five layers of five difference nodes are used. The top and bottom layers are 10 m thick as in the first model, however, the thickness of the middle three layers is adjusted to follow the shape of the hydrogeological layers. The middle layer thickens towards the horizontal model boundaries.



**Figure 11 Models meshes used to test particle tracking in vertically distorted grids**

In both models a source is defined in the top left hand corner and a sink is located in the top right hand corner of the grid. The recharge and discharge rates of the source and sink are identical and are 0.036 m<sup>3</sup>/day per metre horizontal width of the aquifer (in the y-direction or into the paper). The flow models are run to steady-state in order to obtain the nodal flow information required by ZOOPT.

A single particle is tracked through the aquifer using both models. The particle is released from the centre of the node containing the source and is tracked to the discharge point. The particle pathline is defined twice, once using the semi-analytical method and once using the Runge-Kutta technique. These pathlines are shown in Figure 12. In each model the pathline calculated using the Runge-Kutta technique is very similar to that calculated using the semi-analytical method. This indicates that the correction procedures described in Section 3.4 have been incorporated similarly in both methods.



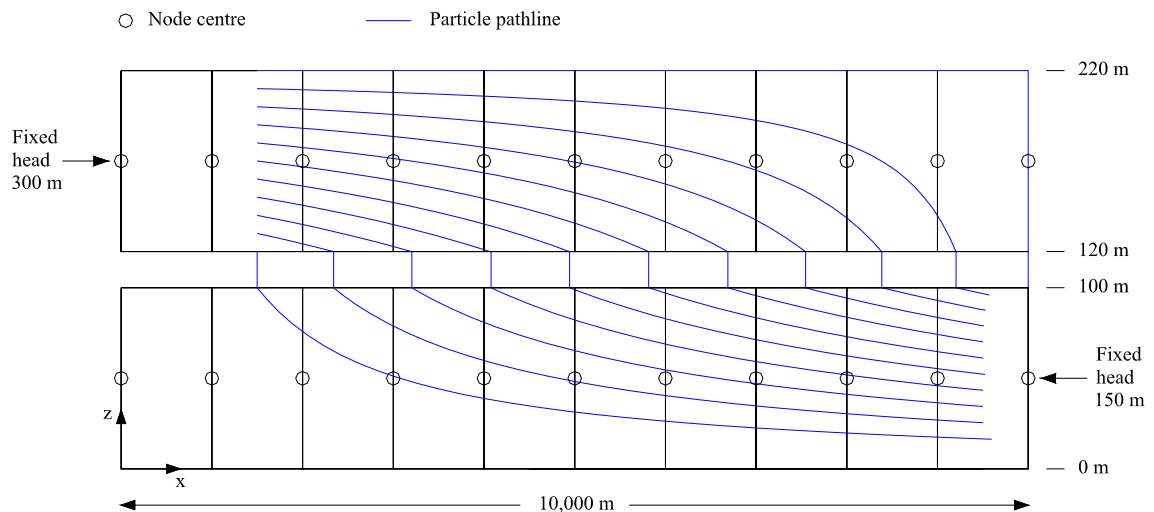
**Figure 12 Comparison between particle paths in the regular orthogonal grid and the vertically distorted mesh models**

Points along pathlines are specified for illustrative purposes only and do not indicate the positions of the particles at end of the tracking steps

The fine grid model, Figure 12a, does not contain vertically distorted grids and consequently does not introduce any additional numerical errors into the particle tracking solution. A comparison of the pathlines generated using the model containing layers of variable thickness (Figure 12b) with the more detailed regular fine grid model indicates that the correction procedures for the two tracking methods are implemented correctly. The pathlines are in adequate agreement. Consequently, it is concluded that the tracking methods operate correctly in vertically distorted mesh models. Additionally, in Figure 12b a particle track is shown (shown in green) for which the correction procedure has not been implemented. This is significantly different from the corrected pathlines (shown in red and blue). The fine grid model indicates that the particle should not enter the lower model layer. However, the uncorrected particle track exits the high hydraulic conductivity middle layer through its base. The major difference between the corrected and uncorrected particle paths illustrates the importance of implementing the correction procedure.

### 5.4 TEST 3: QUASI 3D LAYERING

The assumption is made in ZOOMQ3D and ZOOPT that flow through quasi three-dimensional layers occurs in the vertical direction only. This test simulation is presented to illustrate that ZOOPT tracks particles correctly through these layers. The model grid, shown in Figure 13, represents a cross-section through a confined groundwater system composed of two aquifers separated by a low permeability layer. The low permeability layer is simulated implicitly by adjusting the vertical conductance between the two layers of finite difference nodes. The aquifer is ten kilometres long and each layer of finite difference nodes is 100 m thick. The low permeability layer is 20 m thick. The hydraulic conductivity of the aquifer layers is 10 m/day and the porosity is uniform throughout the model and is 10%.



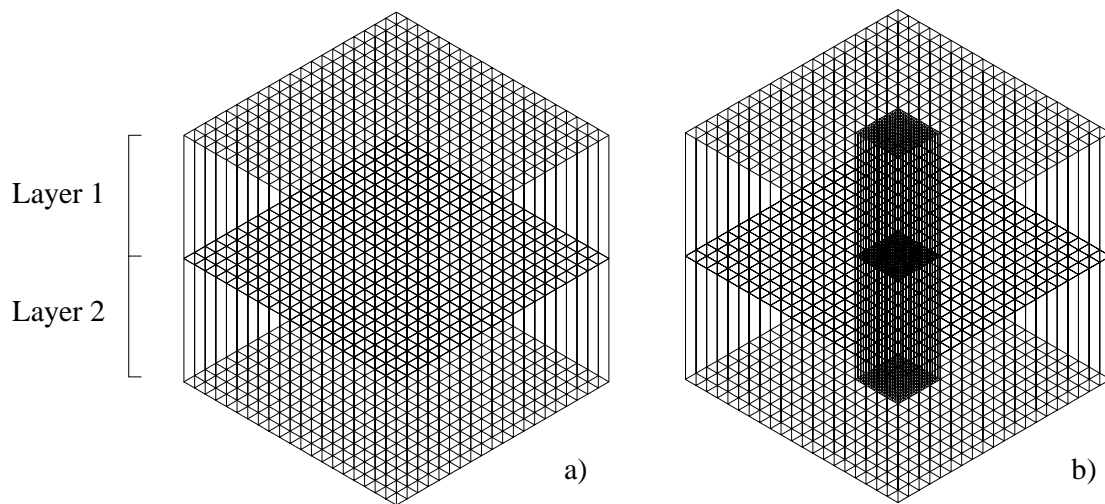
**Figure 13 Model grid showing vertical particle tracks through quasi 3D layer**

The simulation is performed as a steady-state run. Groundwater flows from the fixed head of 300 m in the upper left hand corner of the grid to the fixed head of 150 m in the lower right hand corner of the mesh. Eleven particles are tracked in the forward direction which are released along a vertical line ( $x = 1500$  m) within the upper model layer. The calculated pathlines are shown in Figure 13. These pathlines show that the tracking code correctly moves the particles vertically through the quasi three-dimensional, or low permeability, layer.

## 5.5 TEST 4: FLOW TO A PUMPED WELL

As another visual inspection of particle tracks and as a check on the operation of the tracking routine within locally refined grids, two models are constructed that simulate a pumped well in a two-layer aquifer. Both models are 10 km square and are composed of a horizontal 500 m square base mesh, as shown in Figure 14. The second model's grid, Figure 14b, is refined in the central two kilometre square around the pumped well located at co-ordinate (5000 m, 5000 m). This grid is refined in one-step to a 125 m square mesh. Both layers of the two models are 50 m thick. The aquifer is homogeneous and isotropic and has a hydraulic conductivity of 10 m/day and a porosity of 1%. The pumped well at the centre of the aquifer abstracts water from the lower layer only at a rate of 10 Ml/day. Recharge is distributed uniformly over the aquifer and is applied at a rate of 0.1 mm/day. Consequently, the total recharge is equal to the abstraction.

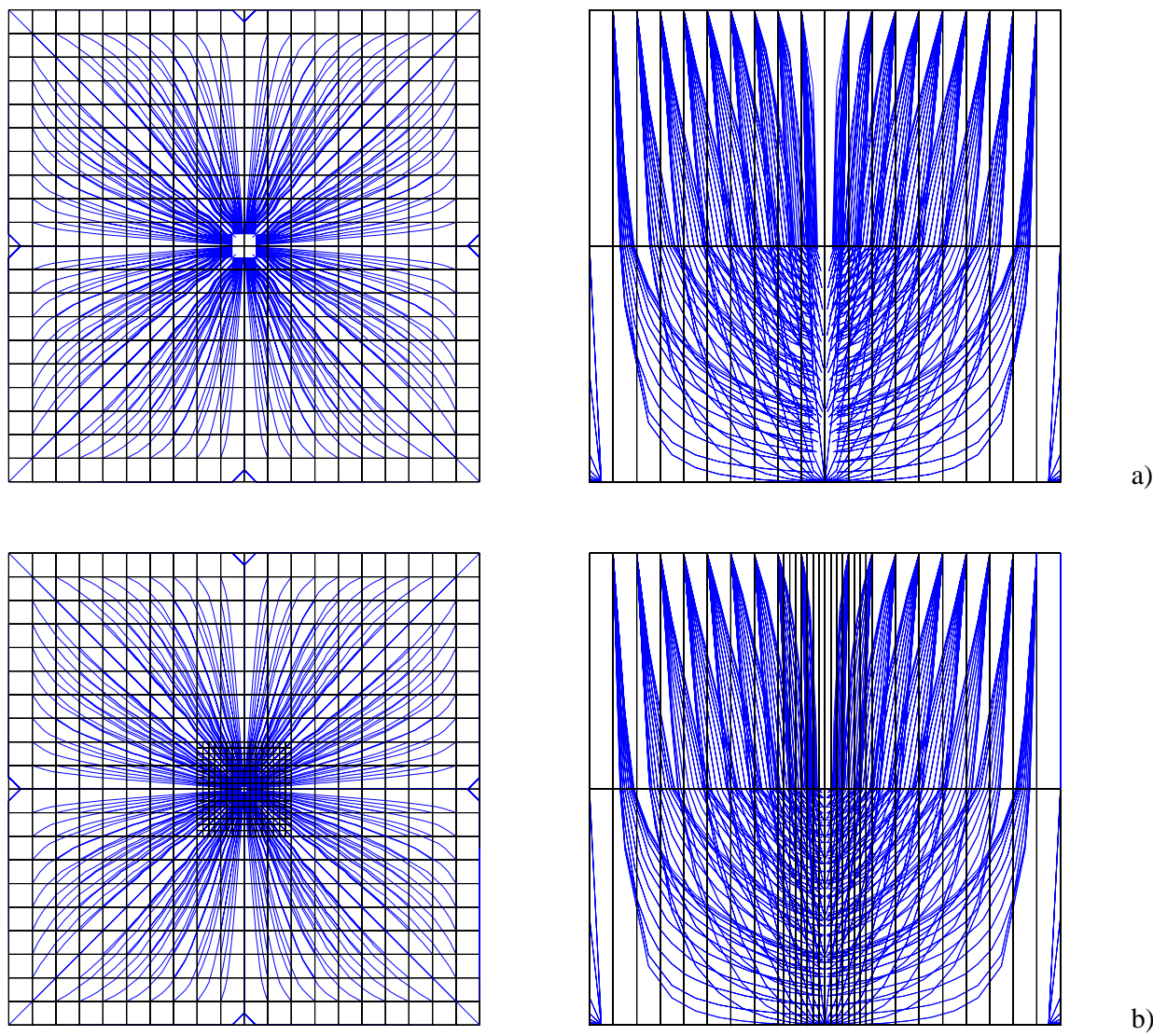
ZOOMQ3D is used to simulate the steady-state flow field and then ZOOPT is used to particle track. Particles are placed at the top of the upper layer at the location of every node in the horizontal direction, that is, every 500 m apart, and then tracked to the well.



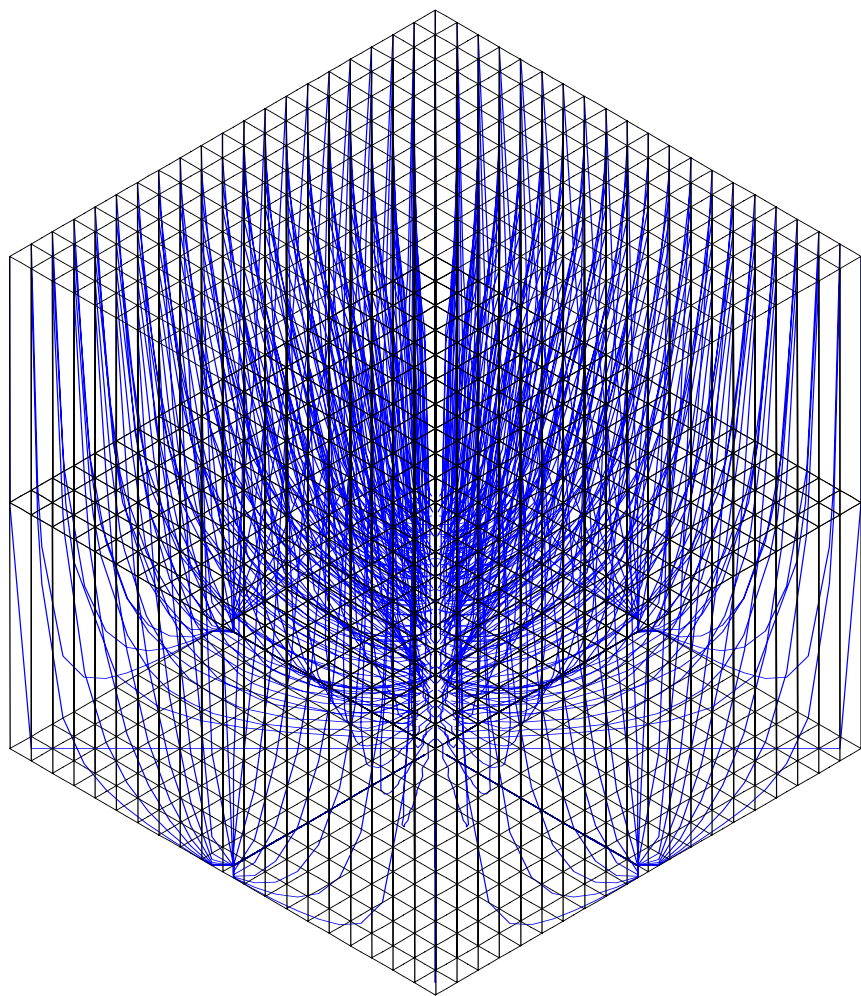
**Figure 14 Model grid used to track particles to a pumped well in a two-layer aquifer**

The pathlines of the particles in the two models are shown in Figure 15 and Figure 16. In Figure 15 the pathlines are drawn in two dimensions in plan and in cross-section. In Figure 16 the pathlines are drawn within a three-dimensional isometric view. The visual inspection of the pathlines indicates that the particle tracking code is operating correctly. For example, particles starting from the corners of the model domain or from the centre of the horizontal boundaries travel in a straight line to the well within the x-y plane. In plan view the pathlines are symmetrical around lines through the well parallel to both the x and y axes as they should be and no particle tracks cross. However, the two models illustrate one of the problems of simulating pumping wells in Cartesian mesh models. In the uniform grid model the pathlines exhibit kinks as they converge towards the well. These are predominantly due to the simulation of radial flow to a well in a Cartesian mesh. Refining the grid around the well produces smoother particle tracks in the neighbourhood of the borehole. Ideally however, when pumped wells are to be simulated in detail, a radial flow model should be used. Finally, the inspection of the pathlines produced by the refined grid model indicates that the particle tracking procedure is suitable in models which contain local grid refinement. This capability will be of benefit when simulating local-scale hydrogeological features in detail.

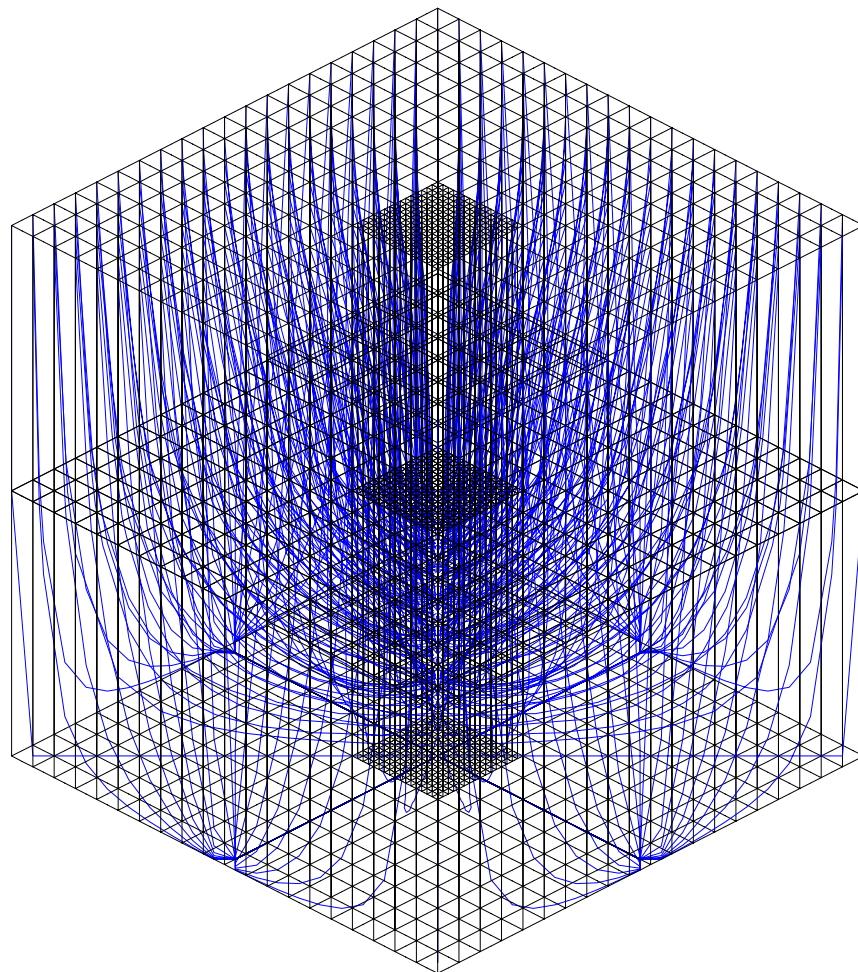




**Figure 15** Particles tracks in plan view and in cross-section for a) the uniform grid and b) the refined grid model



a)



b)

**Figure 16** Particles tracks in 3D isometric view for a) the uniform grid and b) the refined grid model

## 5.6 TEST 5: WEAK SINKS

In Section 3.3 a technique for terminating particles at weak sink nodes induced by abstraction wells is described. This method, developed by Zheng (1994), combines analytical solutions to radial flow and unidirectional regional flow in order to develop a special velocity interpolation scheme around pumping wells. The approach is used to identify which particles enter a well and are terminated and which particles pass through a weak sink node. This example illustrates the benefits of this approach for borehole catchment definition.

A simple one-layer model is constructed which is ten kilometres square with a one kilometre square mesh. The confined aquifer is 100 m thick and has a hydraulic conductivity of 10 m/day and a porosity of 1%. Dirichlet boundary conditions (specified heads) are specified along the left and right hand boundaries. Along the left hand boundary the head is fixed at 200 m above the base of the aquifer and along the right hand boundary the head is specified at 150 m above the aquifer base. A well is located at the centre of the model, which pumps at a constant rate of 10 MI/day during a steady-state simulation.

The steady-state groundwater head profile is used as the starting condition for a time-variant run in which the pumping rate is varied sinusoidally. The specific storage of the aquifer is  $10^{-4}$  and the pumping rates during the four year time-variant simulation are listed in Table 4.

**Table 4 Pumping rates for time-variant simulation in Test 5**

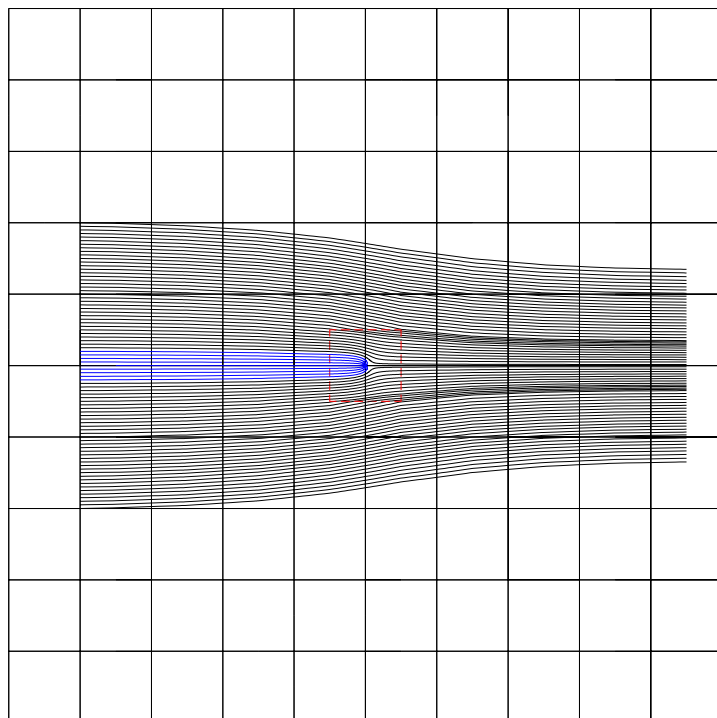
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Year 1	10.0	12.0	14.0	16.0	18.0	20.0	22.0	24.0	22.0	20.0	18.0	16.0
Year 2	14.0	12.0	10.0	8.0	6.0	4.0	2.0	4.0	6.0	8.0	10.0	12.0
Year 3	14.0	16.0	18.0	20.0	22.0	24.0	22.0	20.0	18.0	16.0	14.0	12.0
Year 4	10.0	8.0	6.0	4.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0

As stated in Section 3.1, whilst ZOOPT cannot currently track particles time-variantly, it is straightforward to draw steady-state pathlines using the node-by-node flows terms generated at the end of a specific time step of a time-variant simulation. In Figure 17 such particle tracks are shown at the end of two specific months of the time-variant simulation. In Figure 17a the pathlines are drawn at the end of May in year four, after a month of pumping at a rate of 2.0 MI/day. A weak sink is generated at this time. The borehole catchment (shown in blue) is relatively small because the model calculates that most of the particles that enter the weak sink node pass through it. In Figure 17a most of the particles that enter the cell containing the well, defined by the red square, travel to the eastern model boundary and not to the well. By comparison, in Figure 17b tracks are drawn at the end of April in Year 3, after a month of pumping at a rate of 20 MI/day. The central node is still a weak sink at this time. However, the borehole catchment is significantly larger, as expected, and only two of the particles that enter the weak sink node (red square) pass through it.

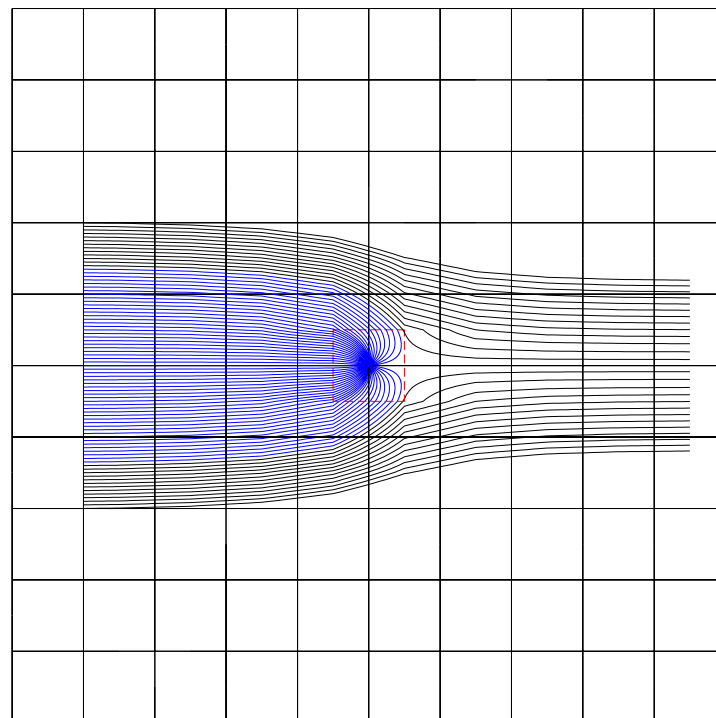
Particles tracking codes that do not solve the problem of weak sink nodes could not be employed in this situation when the objective is to define a borehole catchment. For example, in MODPATH the user can only specify the percentage of particles that should be terminated at the weak sink. In this case, the finite difference grid would have to be refined to ensure that a strong sink is generated so that a borehole catchment can be defined. ZOOPT does not suffer from this problem and can be used to define borehole catchments around weak sink nodes rapidly.



Interface around finite difference node



a)



b)

**Figure 17 Borehole catchments generated around weak sink nodes in Test 5 model using ZOOPT**

## 5.7 TEST 6: PARTICLE TRACKING IN VKD LAYERS

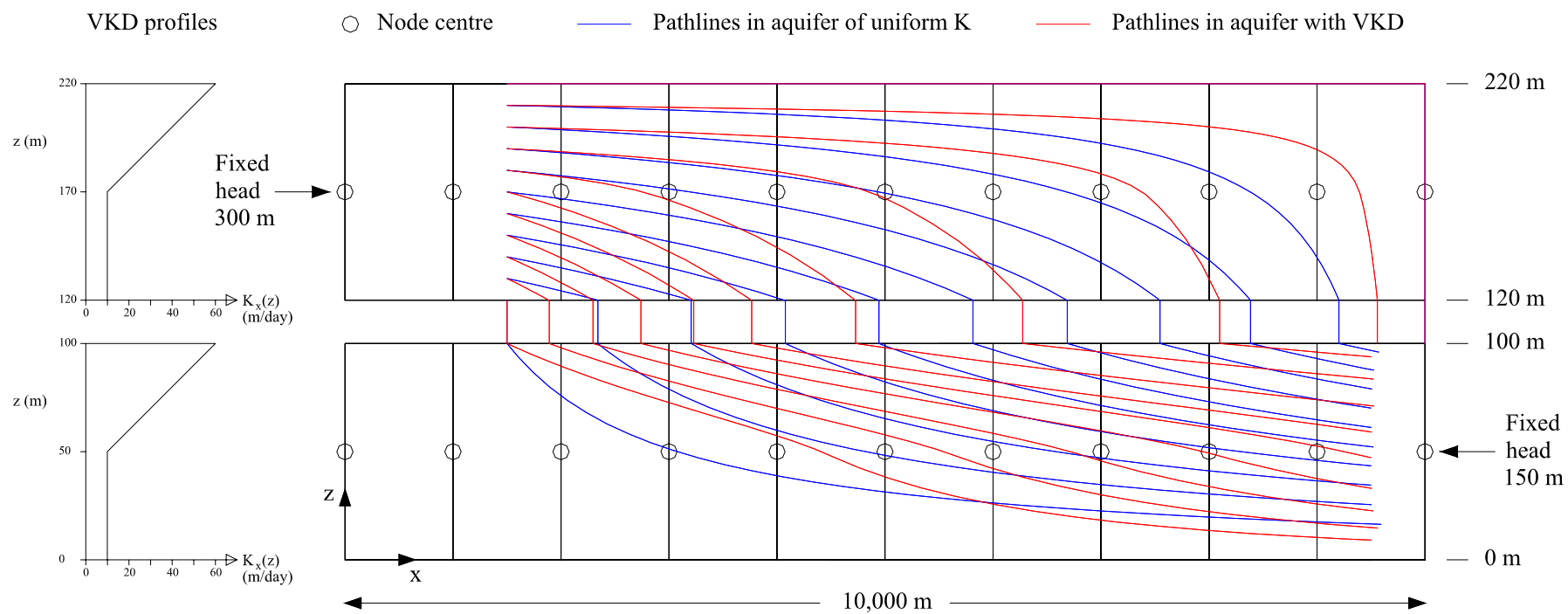
This example is presented to illustrate the application of the particle tracking code to a model in which the horizontal hydraulic conductivity varies with depth. The method by which ZOOPT tracks particles through VKD layers is discussed in Section 3.7. The two models presented here, and shown in Figure 18, are the same as that described in Section 5.4 with the exception of one modification. As previously described, the models represent a cross-section through a confined groundwater system composed of two aquifers separated by a low permeability layer. The vertical conductance between the two layers of finite difference nodes is adjusted to simulate the low permeability layer. The aquifer is ten kilometres long and each layer of finite difference nodes is 100 m thick. The quasi-3D layer is 20 m thick and the porosity is uniform throughout the model and is 10%.

In the model presented in Section 5.4 the hydraulic conductivity is uniform throughout the model domain. The first model used to simulate the system in this test is identical except that hydraulic conductivity is increased to 22.5 m/day in both layers. In the second model, in this example hydraulic conductivity varies with depth as shown in Figure 18. In the lower half of each layer the hydraulic conductivity is uniform and is 10 m/day. Above the vertical midpoint of each layer hydraulic conductivity increases linearly with elevation at a rate of 1 m/day per metre rise in elevation. Consequently, at the top of each layer the horizontal hydraulic conductivity is 60 m/day. By integrating the hydraulic conductivity over the thickness of each layer it is determined that the transmissivity of the layers in each of the two models is the same and is 2250 m<sup>2</sup>/day.

Similarly to the previous model test, the simulation is performed as a steady-state run. Groundwater flows from the fixed head of 300 m in the upper left hand corner of the grid to the fixed head of 150 m in the lower right hand corner of the mesh. Eleven particles are tracked in the forward direction, which are released along a vertical line ( $x = 1500$  m) within the upper model layer. The calculated pathlines for this model containing VKD layers and the previous model in which hydraulic conductivity is uniform are shown in Figure 18.

Because the aquifer is confined and because the transmissivity of each layer in each model is the same, both models calculate the same groundwater heads and inter-nodal flows. It is only the distribution of flow, or, velocity which varies *within* a cell. By examining the travel times of the particles and their associated streamtubes, the groundwater flow pattern could be defined in more detail. However, this would not modify the inter-nodal flows calculated by the groundwater flow model. First, considering the pathlines, those calculated for the VKD model are smooth. It is difficult to validate particle tracking in VKD layers, other than by examining the model code in detail while the program is running. This has been undertaken, however, one indicator of problems associated with particle tracking in VKD layers, was considered to be the generation of irregular pathlines. No erroneous deflections in the particle paths are observed in this test and consequently, it is concluded that the tracking method operates satisfactorily in VKD layers. The influence of the introduction of VKD into the model is easily identified through a comparison of the two sets of pathlines. Particles preferentially travel through the upper half of each model layer where the hydraulic conductivity is greater. However, particles starting from the lower half of the top layer travel downwards more steeply because they are drawn towards the high conductivity zone in the upper half of the bottom layer.

Particle tracking in models containing VKD layers should provide better information with regard to the nature of groundwater flow when, for example, Chalk aquifers are modelled. Whilst increased hydraulic conductivity in the zone of water table fluctuation is generally predominantly associated with the development of fractures, the representation of the system as a continuum of gradually increasing hydraulic conductivity is still a valuable approach. The application of particle tracking in such inhomogeneous aquifers is a development that is of significant benefit.



**Figure 18 Comparison of particle paths in a uniform hydraulic conductivity and a VKD quasi two-layer model**

## 6 Conclusions and recommendations

The semi-analytical and Runge-Kutta methods have been developed within the particle tracking code, ZOOPT, to define pathlines through groundwater flow fields calculated using the regional groundwater model, ZOOMQ3D. Except in the vicinity of some specific model mechanisms, the velocity of a particle is calculated by linearly interpolating the velocities on the six interfaces surrounding the corresponding finite difference node. This enables the application of the semi-analytical technique (Pollock, 1988), which is computationally efficient.

The Runge-Kutta method is used to track particles in models, which incorporate the vertical variation of hydraulic conductivity with depth (VKD). In this case the horizontal velocity depends on the elevation of the particle and the semi-analytical method cannot be used. The capability to define particle pathlines in VKD models is expected to be a significant benefit when investigating groundwater flow patterns in Chalk aquifers, for example. The author is unaware of any other groundwater flow model/particle tracking code that has this capability.

The particle tracking code, ZOOPT, does not suffer from the problems associated with ‘weak’ sink nodes. The application of the special velocity interpolation scheme, developed by Zheng (1994), facilitates the definition of borehole catchments around weak sinks. This is an improvement over MODPATH (Pollock, 1994), a widely applied particle tracking code, in which the user selects the percentage of particles to terminate at a weak sink.

In addition to these capabilities, ZOOPT can track particles in both the forward and reverse directions. It correctly moves particles through vertically distorted model layers and quasi three-dimensional models and can define pathlines through areas of local grid refinement.

Whilst the particle tracking code incorporates significant functionality, additional capabilities should be added in a second stage of development. Currently, the model cannot track particles for a specific length of time, for example, to define a fifty-day capture zone for an abstraction well. This is a useful capability and would take little time to incorporate in ZOOPT. Secondly, the incorporation of the ability to track particles time-variantly is important, but ZOOPT cannot yet do this. However, it is designed so that the time-variant particle tracking technique of Lu (1994), in which a particle’s velocity is interpolated linearly in time as well as in space, can be incorporated easily. More conventional time-variant particle tracking techniques would take less time to develop.

## 7 References

- BEAR, J. (1979). Hydraulics of groundwater. McGraw-Hill, New York.
- BLANDFORD, T.N. AND HUYAKORN, P.S. (1991). WHPA: A modular semi-analytical model for the delineation of wellhead protection areas, version 2.0. Report to the U.S. Environmental Protection Agency, Washington, DC.
- FARAHMAND-RAZAVI, A. (1995). Theory and practice of contaminant transport modelling using the random walk. Unpublished Ph.D. Thesis, University of Birmingham.
- FRANZ, T. AND GUIGER, N. (1990). FLOWPATH, two-dimensional horizontal aquifer simulation model. Waterloo Hydrogeologic Software, Waterloo, Ontario.
- GOODE, D.J. (1990). Particle velocity interpolation in block-centred finite difference groundwater flow models. Water Resources Research, Vol. 26, No. 5, pp 925-940.
- JACKSON, C.R. (2001). The development and validation the object-oriented quasi three-dimensional regional groundwater model ZOOMQ3D. British Geological Survey Internal Report IR/01/144.
- JACKSON, C.R. (2002). The representation of the variation of hydraulic conductivity with depth in the object-oriented regional groundwater model ZOOMQ3D. British Geological Survey Commissioned Report CR/02/152N.
- KONIKOW, L.F. AND BREDEHOEFT, J.D. (1978). Computer model of two-dimensional solute transport and dispersion in ground water. USGS Water Resources Investigations, Book 7, Chapter C2.
- LU, N. (1994). A semi-analytical method of path line computation for transient finite-difference groundwater flow models. Water Resources Research. Vol. 30, No. 8, 2449-2459.
- MCDONALD, M.G., AND HARBAUGH, A.W. (1988). A modular three-dimensional finite-difference ground-water flow model. U.S. Geological Survey, Techniques of Water Resources Investigations. Book 6, Chapter A1.
- POLLOCK, D.W. (1988). Semianalytical computation of path lines for finite-difference models. Ground Water, Vol. 26, No. 6.
- POLLOCK, D.W. (1989). Documentation of computer programs to compute and display pathlines using results from the U.S. Geological Survey modular three-dimensional finite-difference ground-water model. U.S. Geological Survey Open-File report 89-381.
- POLLOCK, D.W. (1994). User's Guide for MODPATH/MODPATH-PLOT, Version 3: A particle tracking post-processing package for MODFLOW, the U.S. Geological Survey finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 94-464.
- PRICKETT, T.A., NAYMIK, T.G. AND LONNQUIST, C.G. (1981). A random walk solute transport model for selected groundwater quality evaluations. Bulletin 65. Illinois State Water Survey.
- SHAFER, J.M. (1987). GWPATh: interactive ground-water flow path analysis. Bulletin 69, Illinois State Water Survey, Champaign, IL.
- TAYLOR, A., HULME, P., HUGHES, A. AND RUSHTON, K.R. (2001). Representation of variable hydraulic conductivity with depth in MODFLOW. MODFLOW 2001 and Other Modeling Odysseys – Conference Proceedings, eds. Seo, Poeter, Zheng and Poeter.
- ZHENG, C. (1989). PATH3D, a groundwater path and travel-time simulator, version 3.0 user's manual. S.S. Papadopolous & Associates, Inc. Bethesda, MD.
- ZHENG, C. (1990). MT3D: A modular three-dimensional transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems. Report to the U.S. E.P.A., Ada, OK.
- ZHENG, C. (1994). Analysis of particle tracking errors associated with spatial discretisation. Ground Water, Vol. 32, No. 5, pp 821-828.
- ZHENG, C. AND BENNETT, G.D. (1995). Applied contaminant transport modelling. Van Nostrand Reinhold, New York.